

**File: ./.gitignore**

```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
pip-wheel-metadata/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST

# PyInstaller
# Usually these files are written by a python script from a template
# before PyInstaller builds the exe, so as to inject date/other infos into it.
*.manifest
*.spec

# Installer logs
pip-log.txt
pip-delete-this-directory.txt

# Unit test / coverage reports
htmlcov/
.tox/
.nox/
.coverage
.coverage.*
.cache
nosetests.xml
coverage.xml
*.cover
*.py,cover
.hypothesis/
.pytest_cache/

# Translations
*.mo
*.pot

# Django stuff:
*.log
local_settings.py
db.sqlite3
db.sqlite3-journal

# Flask stuff:
instance/
.webassets-cache

# Scrapy stuff:
.scrapy

# Sphinx documentation
docs/_build/

# PyBuilder
target/

# Jupyter Notebook
.ipynb_checkpoints

# IPython
profile_default/
ipython_config.py

# pyenv
.python-version

# pipenv
# According to pypa/pipenv#598, it is recommended to include Pipfile.lock in version control.
# However, in case of collaboration, if having platform-specific dependencies or dependencies
# having no cross-platform support, pipenv may install dependencies that don't work, or not
# install all needed dependencies.
#Pipfile.lock

# PEP 582; used by e.g. github.com/David-OConnor/pyflow
__pypackages__/

# Celery stuff
celerybeat-schedule
celerybeat.pid

# SageMath parsed files
```

```
*.sage.py

# Environments
.env
.venv
env/
venv/
ENV/
env.bak/
venv.bak/

# Spyder project settings
.spyderproject
.spyproject

# Rope project settings
.ropeproject

# makedocs documentation
/site

# mypy
.mypy_cache/
.dmypy.json
dmypy.json

# Pyre type checker
.pyre/

images/
```

**File: ./README.md**

```
1 # health-cloud-back-end
```

**File: ./git/index**

```
[binary]
```

**File: ./git/HEAD**

```
ref: refs/heads/main
```

**File: ./git/packed-refs**

```
# pack-refs with: peeled fully-peeled sorted
fccad7438134b2cfd5399517db0e30bddb4591f refs/remotes/origin/main
```

**File: ./git/config**

```
[core]
repositoryformatversion = 0
filemode = true
bare = false
logallrefupdates = true
[remote "origin"]
url = https://github.com/gabrielsm0405/health-cloud-back-end.git
fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
remote = origin
merge = refs/heads/main
```

**File: ./git/description**

Unnamed repository; edit this file 'description' to name the repository.

**File: ./git/hooks/fsmonitor-watchman.sample**

```
#!/usr/bin/perl

use strict;
use warnings;
use IPC::Open2;

# An example hook script to integrate Watchman
# (https://facebook.github.io/watchman/) with git to speed up detecting
# new and modified files.
#
# The hook is passed a version (currently 1) and a time in nanoseconds
# formatted as a string and outputs to stdout all files that have been
# modified since the given time. Paths must be relative to the root of
# the working tree and separated by a single NUL.
#
# To enable this hook, rename this file to "query-watchman" and set
# 'git config core.fsmonitor .git/hooks/query-watchman'
#
my ($version, $time) = @ARGV;

# Check the hook interface version

if ($version == 1) {
```

```

# convert nanoseconds to seconds
# subtract one second to make sure watchman will return all changes
$time = int ($time / 1000000000) - 1;
} else {
die "Unsupported query-fsmonitor hook version '$version'.\n" .
"Falling back to scanning...\n";
}

my $git_work_tree;
if ($^O =~ 'msys' || $^O =~ 'cygwin') {
$git_work_tree = Win32::GetCwd();
$git_work_tree =~ tr/\V//;
} else {
require Cwd;
$git_work_tree = Cwd::cwd();
}

my $retry = 1;

launch_watchman();

sub launch_watchman {

my $pid = open2(\*CHLD_OUT, \*CHLD_IN, 'watchman -j --no-pretty')
or die "open2() failed: $!\n" .
"Falling back to scanning...\n";

# In the query expression below we're asking for names of files that
# changed since $time but were not transient (ie created after
# $time but no longer exist).
#
# To accomplish this, we're using the "since" generator to use the
# recency index to select candidate nodes and "fields" to limit the
# output to file names only.

my $query = <<" END";
["query", "$git_work_tree", {
"since": $time,
"fields": ["name"]
}]
END

print CHLD_IN $query;
close CHLD_IN;
my $response = do {local $/; <CHLD_OUT>};

die "Watchman: command returned no output.\n" .
"Falling back to scanning...\n" if $response eq "";
die "Watchman: command returned invalid output: $response\n" .
"Falling back to scanning...\n" unless $response =~ /^{/;

my $json_pkg;
eval {
require JSON::XS;
$json_pkg = "JSON::XS";
1;
} or do {
require JSON::PP;
$json_pkg = "JSON::PP";
};

my $o = $json_pkg->new->utf8->decode($response);

if ($retry > 0 and $o->{error} and $o->{error} =~ m/unable to resolve root .* directory (.*) is not watched/) {
print STDERR "Adding '$git_work_tree' to watchman's watch list.\n";
$retry--;
qx/watchman watch "$git_work_tree"/;
die "Failed to make watchman watch '$git_work_tree'.\n" .
"Falling back to scanning...\n" if $? != 0;

# Watchman will always return all files on the first query so
# return the fast "everything is dirty" flag to git and do the
# Watchman query just to get it over with now so we won't pay
# the cost in git to look up each individual file.
print "\0";
eval { launch_watchman() };
exit 0;
}

die "Watchman: $o->{error}.\n" .
"Falling back to scanning...\n" if $o->{error};

binmode STDOUT, ":utf8";
local $, = "\0";
print @{$o->{files}};
}

```

**File: `./git/hooks/update.sample`**

```

#!/bin/sh
#
# An example hook script to block unannotated tags from entering.
# Called by "git receive-pack" with arguments: refname sha1-old sha1-new
#
# To enable this hook, rename this file to "update".
#
# Config
# -----
# hooks.allowunannotated
# This boolean sets whether unannotated tags will be allowed into the
# repository. By default they won't be.

```

```
# hooks.allowdeletetag
# This boolean sets whether deleting tags will be allowed in the
# repository. By default they won't be.
# hooks.allowmodifytag
# This boolean sets whether a tag may be modified after creation. By default
# it won't be.
# hooks.allowdeletebranch
# This boolean sets whether deleting branches will be allowed in the
# repository. By default they won't be.
# hooks.denycreatebranch
# This boolean sets whether remotely creating branches will be denied
# in the repository. By default this is allowed.
#

# --- Command line
refname="$1"
oldrev="$2"
newrev="$3"

# --- Safety check
if [ -z "$GIT_DIR" ]; then
echo "Don't run this script from the command line." >&2
echo " (if you want, you could supply GIT_DIR then run" >&2
echo " $0 <ref> <oldrev> <newrev>)" >&2
exit 1
fi

if [ -z "$refname" -o -z "$oldrev" -o -z "$newrev" ]; then
echo "usage: $0 <ref> <oldrev> <newrev>" >&2
exit 1
fi

# --- Config
allowunannotated=$(git config --bool hooks.allowunannotated)
allowdeletebranch=$(git config --bool hooks.allowdeletebranch)
denycreatebranch=$(git config --bool hooks.denycreatebranch)
allowdeletetag=$(git config --bool hooks.allowdeletetag)
allowmodifytag=$(git config --bool hooks.allowmodifytag)

# check for no description
projectdesc=$(sed -e '1q' "$GIT_DIR/description")
case "$projectdesc" in
"Unnamed repository"*)
echo "Project description file hasn't been set" >&2
exit 1
;;
esac

# --- Check types
# if $newrev is 0000...0000, it's a commit to delete a ref.
zero="0000000000000000000000000000000000000000000000000000000000000000"
if [ "$newrev" = "$zero" ]; then
newrev_type=delete
else
newrev_type=$(git cat-file -t $newrev)
fi

case "$refname","$newrev_type" in
refs/tags/*,commit)
# un-annotated tag
short_refname=${refname##refs/tags/}
if [ "$allowunannotated" != "true" ]; then
echo "The un-annotated tag, $short_refname, is not allowed in this repository" >&2
echo "Use 'git tag -a -s' for tags you want to propagate." >&2
exit 1
fi
;;
refs/tags/*,delete)
# delete tag
if [ "$allowdeletetag" != "true" ]; then
echo "Deleting a tag is not allowed in this repository" >&2
exit 1
fi
;;
refs/tags/*,tag)
# annotated tag
if [ "$allowmodifytag" != "true" ] && git rev-parse $refname > /dev/null 2>&1
then
echo "Tag '$refname' already exists." >&2
echo "Modifying a tag is not allowed in this repository." >&2
exit 1
fi
;;
refs/heads/*,commit)
# branch
if [ "$oldrev" = "$zero" -a "$denycreatebranch" = "true" ]; then
echo "Creating a branch is not allowed in this repository" >&2
exit 1
fi
;;
refs/heads/*,delete)
# delete branch
if [ "$allowdeletebranch" != "true" ]; then
echo "Deleting a branch is not allowed in this repository" >&2
exit 1
fi
;;
refs/remotes/*,commit)
# tracking branch
;;
refs/remotes/*,delete)
# delete tracking branch
if [ "$allowdeletebranch" != "true" ]; then

```

```

echo "*** Deleting a tracking branch is not allowed in this repository" >&2
exit 1
fi
;;
*)
# Anything else (is there anything else?)
echo "*** Update hook: unknown type of update to ref $refname of type $newrev_type" >&2
exit 1
;;
esac

# --- Finished
exit 0

```

**File: `./git/hooks/pre-applypatch.sample`**

```

#!/bin/sh
#
# An example hook script to verify what is about to be committed
# by applypatch from an e-mail message.
#
# The hook should exit with non-zero status after issuing an
# appropriate message if it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-applypatch".

. git-sh-setup
precommit="$(git rev-parse --git-path hooks/pre-commit)"
test -x "$precommit" && exec "$precommit" ${1+"$@"}
:

```

**File: `./git/hooks/pre-push.sample`**

```

#!/bin/sh

# An example hook script to verify what is about to be pushed. Called by "git
# push" after it has checked the remote status, but before anything has been
# pushed. If this script exits with a non-zero status nothing will be pushed.
#
# This hook is called with the following parameters:
#
# $1 -- Name of the remote to which the push is being done
# $2 -- URL to which the push is being done
#
# If pushing without using a named remote those arguments will be equal.
#
# Information about the commits which are being pushed is supplied as lines to
# the standard input in the form:
#
# <local ref> <local sha1> <remote ref> <remote sha1>
#
# This sample shows how to prevent push of commits where the log message starts
# with "WIP" (work in progress).

remote="$1"
url="$2"

z40=0000000000000000000000000000000000000000000000000000000000000000

while read local_ref local_sha remote_ref remote_sha
do
if [ "$local_sha" = $z40 ]
then
# Handle delete
:
else
if [ "$remote_sha" = $z40 ]
then
# New branch, examine all commits
range="$local_sha"
else
# Update to existing branch, examine new commits
range="$remote_sha..$local_sha"
fi

# Check for WIP commit
commit=`git rev-list -n 1 --grep '^WIP' "$range"`
if [ -n "$commit" ]
then
echo >&2 "Found WIP commit in $local_ref, not pushing"
exit 1
fi
done

exit 0

```

**File: `./git/hooks/pre-receive.sample`**

```

#!/bin/sh
#
# An example hook script to make use of push options.
# The example simply echoes all push options that start with 'echoback='
# and rejects all pushes when the "reject" push option is used.
#
# To enable this hook, rename this file to "pre-receive".

if test -n "$GIT_PUSH_OPTION_COUNT"

```

```

then
i=0
while test "$i" -lt "$GIT_PUSH_OPTION_COUNT"
do
eval "value=\$GIT_PUSH_OPTION_$i"
case "$value" in
echohack=*)
echo "echo from the pre-receive-hook: ${value#*=}" >&2
;;
reject)
exit 1
esac
i=$((i + 1))
done
fi

```

**File: `./git/hooks/pre-merge-commit.sample`**

```

#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git merge" with no arguments. The hook should
# exit with non-zero status after issuing an appropriate message to
# stderr if it wants to stop the merge commit.
#
# To enable this hook, rename this file to "pre-merge-commit".

. git-sh-setup
test -x "$GIT_DIR/hooks/pre-commit" &&
exec "$GIT_DIR/hooks/pre-commit"
:

```

**File: `./git/hooks/applypatch-msg.sample`**

```

#!/bin/sh
#
# An example hook script to check the commit log message taken by
# applypatch from an e-mail message.
#
# The hook should exit with non-zero status after issuing an
# appropriate message if it wants to stop the commit. The hook is
# allowed to edit the commit message file.
#
# To enable this hook, rename this file to "applypatch-msg".

. git-sh-setup
commitmsg="$(git rev-parse --git-path hooks/commit-msg)"
test -x "$commitmsg" && exec "$commitmsg" ${1+"$@"}
:

```

**File: `./git/hooks/pre-commit.sample`**

```

#!/bin/sh
#
# An example hook script to verify what is about to be committed.
# Called by "git commit" with no arguments. The hook should
# exit with non-zero status after issuing an appropriate message if
# it wants to stop the commit.
#
# To enable this hook, rename this file to "pre-commit".

if git rev-parse --verify HEAD >/dev/null 2>&1
then
against=HEAD
else
# Initial commit: diff against an empty tree object
against=$(git hash-object -t tree /dev/null)
fi

# If you want to allow non-ASCII filenames set this variable to true.
allownonascii=$(git config --bool hooks.allownonascii)

# Redirect output to stderr.
exec 1>&2

# Cross platform projects tend to avoid non-ASCII filenames; prevent
# them from being added to the repository. We exploit the fact that the
# printable range starts at the space character and ends with tilde.
if [ "$allownonascii" != "true" ] &&
# Note that the use of brackets around a tr range is ok here, (it's
# even required, for portability to Solaris 10's /usr/bin/tr), since
# the square bracket bytes happen to fall in the designated range.
test $(git diff --cached --name-only --diff-filter=A -z $against |
LC_ALL=C tr -d '[' ~]0' | wc -c) != 0
then
cat <<\EOF
Error: Attempt to add a non-ASCII file name.

```

This can cause problems if you want to work with people on other platforms.

To be portable it is advisable to rename the file.

If you know what you are doing you can disable this check using:

```

git config hooks.allownonascii true
EOF
exit 1
fi

```

```
# If there are whitespace errors, print the offending file names and fail.
exec git diff-index --check --cached $against --
```

**File: `./git/hooks/prepare-commit-msg.sample`**

```
#!/bin/sh
#
# An example hook script to prepare the commit log message.
# Called by "git commit" with the name of the file that has the
# commit message, followed by the description of the commit
# message's source. The hook's purpose is to edit the commit
# message file. If the hook fails with a non-zero status,
# the commit is aborted.
#
# To enable this hook, rename this file to "prepare-commit-msg".

# This hook includes three examples. The first one removes the
# "# Please enter the commit message..." help message.
#
# The second includes the output of "git diff --name-status -r"
# into the message, just before the "git status" output. It is
# commented because it doesn't cope with --amend or with squashed
# commits.
#
# The third example adds a Signed-off-by line to the message, that can
# still be edited. This is rarely a good idea.

COMMIT_MSG_FILE=$1
COMMIT_SOURCE=$2
SHA1=$3

/usr/bin/perl -i.bak -ne 'print unless(m/^\. Please enter the commit message/..m/^#$/)' "$COMMIT_MSG_FILE"

# case "$COMMIT_SOURCE,$SHA1" in
# ,|template,)
# /usr/bin/perl -i.bak -pe '
# print "\n" . `git diff --cached --name-status -r`
# if /^#/ && $first++ == 0' "$COMMIT_MSG_FILE" ;;
# *) ;;
# esac

# SOB=$(git var GIT_COMMITTER_IDENT | sed -n 's/^(\.*)> \1/p')
# git interpret-trailers --in-place --trailer "$SOB" "$COMMIT_MSG_FILE"
# if test -z "$COMMIT_SOURCE"
# then
# /usr/bin/perl -i.bak -pe 'print "\n" if !$first_line++' "$COMMIT_MSG_FILE"
# fi
```

**File: `./git/hooks/commit-msg.sample`**

```
#!/bin/sh
#
# An example hook script to check the commit log message.
# Called by "git commit" with one argument, the name of the file
# that has the commit message. The hook should exit with non-zero
# status after issuing an appropriate message if it wants to stop the
# commit. The hook is allowed to edit the commit message file.
#
# To enable this hook, rename this file to "commit-msg".

# Uncomment the below to add a Signed-off-by line to the message.
# Doing this in a hook is a bad idea in general, but the prepare-commit-msg
# hook is more suited to it.
#
# SOB=$(git var GIT_AUTHOR_IDENT | sed -n 's/^(\.*)> \1/p')
# grep -qs "^$SOB" "$1" || echo "$SOB" >> "$1"

# This example catches duplicate Signed-off-by lines.

test "" = "$(grep '^Signed-off-by: ' "$1" |
sort | uniq -c | sed -e '/^[ ]*1[ ]/d')" || {
echo >&2 Duplicate Signed-off-by lines.
exit 1
}
```

**File: `./git/hooks/post-update.sample`**

```
#!/bin/sh
#
# An example hook script to prepare a packed repository for use over
# dumb transports.
#
# To enable this hook, rename this file to "post-update".

exec git update-server-info
```

**File: `./git/hooks/pre-rebase.sample`**

```
#!/bin/sh
#
# Copyright (c) 2006, 2008 Junio C Hamano
#
# The "pre-rebase" hook is run just before "git rebase" starts doing
# its job, and can prevent the command from running by exiting with
# non-zero status.
```

```
#
# The hook is called with the following parameters:
#
# $1 -- the upstream the series was forked from.
# $2 -- the branch being rebased (or empty when rebasing the current branch).
#
# This sample shows how to prevent topic branches that are already
# merged to 'next' branch from getting rebased, because allowing it
# would result in rebasing already published history.
```

```
publish=next
basebranch="$1"
if test "$#" = 2
then
topic="refs/heads/$2"
else
topic=`git symbolic-ref HEAD` ||
exit 0 ;# we do not interrupt rebasing detached HEAD
fi
```

```
case "$topic" in
refs/heads/??/*)
;;
*)
exit 0 ;# we do not interrupt others.
;;
esac
```

```
# Now we are dealing with a topic branch being rebased
# on top of master. Is it OK to rebase it?
```

```
# Does the topic really exist?
git show-ref -q "$topic" || {
echo >&2 "No such branch $topic"
exit 1
}
```

```
# Is topic fully merged to master?
not_in_master=`git rev-list --pretty=oneline ^master "$topic"`
if test -z "$not_in_master"
then
echo >&2 "$topic is fully merged to master; better remove it."
exit 1 ;# we could allow it, but there is no point.
fi
```

```
# Is topic ever merged to next? If so you should not be rebasing it.
only_next_1=`git rev-list ^master "^$topic" ${publish} | sort`
only_next_2=`git rev-list ^master ${publish} | sort`
if test "$only_next_1" = "$only_next_2"
then
not_in_topic=`git rev-list "^$topic" master`
if test -z "$not_in_topic"
then
echo >&2 "$topic is already up to date with master"
exit 1 ;# we could allow it, but there is no point.
else
exit 0
fi
else
not_in_next=`git rev-list --pretty=oneline ^${publish} "$topic"`
/usr/bin/perl -e '
my $topic = $ARGV[0];
my $msg = "* $topic has commits already merged to public branch:\n";
my (%not_in_next) = map {
/^[0-9a-f]+/ };
($1 => 1);
} split(/\n/, $ARGV[1]);
for my $elem (map {
/^[0-9a-f]+/ } (.*$);
[$1 => $2];
} split(/\n/, $ARGV[2])) {
if (!exists $not_in_next{$elem->[0]}) {
if ($msg) {
print STDERR $msg;
undef $msg;
}
print STDERR " $elem->[1]\n";
}
}
' "$topic" "$not_in_next" "$not_in_master"
exit 1
fi
```

<<\DOC\_END

This sample hook safeguards topic branches that have been published from being rewound.

The workflow assumed here is:

- \* Once a topic branch forks from "master", "master" is never merged into it again (either directly or indirectly).

- \* Once a topic branch is fully cooked and merged into "master", it is deleted. If you need to build on top of it to correct earlier mistakes, a new topic branch is created by forking at the tip of the "master". This is not strictly necessary, but it makes it easier to keep your history simple.

- \* Whenever you need to test or publish your changes to topic branches, merge them into "next" branch.

The script, being an example, hardcodes the publish branch name



With this workflow, you would want to know:

(2) ... if a topic branch has been fully merged to "master". Then you can delete it. More importantly, you should not build on top of it – other people may already want to change things related to the topic as patches against your "master", so if you need further changes, it is better to fork the topic (perhaps with the same name) afresh from the tip of "master".

```

0---0---0---0---0---0---0---0---0---0 "next"
////
/a---a---b A //
////
// C---C---C B /
///\ \ /
///b---b C \ /
///\ \ /
---0---0---0---0---0---0---0---0---0---0 "master"

```

DOC\_END

[binary]

[binary]

```
# git ls-files --others --exclude-from=.git/info/exclude
# Lines that start with '#' are comments.
# For a project mostly in C, the following would be a good set of
# exclude patterns (uncomment them if you want to use them):
# *.o[ad]
# *
```

```
ref: refs/remotes/origin/main
```

fccad7438134b2cfd5399517db0e30bdddb4591f

File: ../.git/logs/refs/remotes/origin/HEAD

```
00000000000000000000000000000000 fccad7438134b2cfd5399517db0e30bddd4591f Gabriel Marques <gabrielism0405@gmail.com>  
1644591723 -0300 clone: from https://github.com/gabrielism0405/health-cloud-back-end.git
```

```
00000000000000000000000000000000 fccad7438134b2cfd5399517db0e30bddd4591f Gabriel Marques <gabrielism0405@gmail.com>  
1644591723 -0300 clone: from https://github.com/gabrielism0405/health-cloud-back-end.git
```

```

1 """health_cloud_back_end URL Configuration
2
3 The 'urlpatterns' list routes URLs to views. For more information please see:
4 https://docs.djangoproject.com/en/1.11/topics/http/urls/
5 Examples:
6 Function views
7     1. Add an import:  from my_app import views
8     2. Add a URL to urlpatterns:  url(r'^$', views.home, name='home')
9 Class-based views
10    1. Add an import:  from other_app.views import Home
11    2. Add a URL to urlpatterns:  url(r'^$', Home.as_view(), name='home')
12 Including another URLconf
13    1. Import the include() function: from django.conf.urls import url, include
14    2. Add a URL to urlpatterns:  url(r'^blog/', include('blog.urls'))
15 """
16 from django.urls import path, include
17 from django.contrib import admin
18 from health_cloud_back_end.app import views
19 from django.conf.urls.static import static
20 import os
21 from .settings import BASE_DIR
22
23 urlpatterns = [
24     path('admin/', admin.site.urls),
25     path('inference/', views.InferenceView.as_view({'post': 'makeInference'})),
26 ] + static("images/", document_root=os.path.join(BASE_DIR, "images")) + static("/static/" document_root=os.path.join(BASE_DIR, 'static'))

```

```

1  #-*- coding: utf-8 -*-
2  from __future__ import unicode_literals
3
4  from django.contrib import admin
5  from .models import Results, Keys, HeatmapLink
6
7  # Register your models here.
8
9  class ResultsAdmin(admin.ModelAdmin):
10     fields = ('id', 'key', 'originalImage', 'heatmaps_links', 'created_at')
11     readonly_fields = ('id', 'originalImage', 'heatmaps_links', 'created_at')
12     actions = ('really_delete_selected',)
13
14     def get_actions(self, request):
15         actions = super(ResultsAdmin, self).get_actions(request)
16         del actions['delete_selected']
17         return actions
18
19     def really_delete_selected(self, request, queryset):
20         for obj in queryset:
21             obj.delete()
22
23         if queryset.count() == 1:
24             message_bit = "1 result entry was"
25         else:
26             message_bit = "%s result entries were" % queryset.count()
27         self.message_user(request, "%s successfully deleted." % message_bit)
28     really_delete_selected.short_description = "Delete selected entries"
29
30 class KeysAdmin(admin.ModelAdmin):
31     fields = ('id', 'key', 'counter', 'created_at')
32     readonly_fields = ('id', 'key', 'counter', 'created_at')
33     actions = ('really_delete_selected',)
34
35     def get_actions(self, request):
36         actions = super(KeysAdmin, self).get_actions(request)
37         del actions['delete_selected']
38         return actions
39
40     def really_delete_selected(self, request, queryset):
41         for obj in queryset:
42             obj.delete()
43
44         if queryset.count() == 1:
45             message_bit = "1 key entry was"
46         else:
47             message_bit = "%s key entries were" % queryset.count()
48         self.message_user(request, "%s successfully deleted." % message_bit)
49     really_delete_selected.short_description = "Delete selected entries"
50
51 class HeatmapsLinkAdmin(admin.ModelAdmin):
52     fields = ('id', 'pathology', 'link')
53     readonly_fields = ('id', 'pathology', 'link')
54     actions = ('really_delete_selected',)
55
56     def get_actions(self, request):
57         actions = super(HeatmapsLinkAdmin, self).get_actions(request)
58         del actions['delete_selected']
59         return actions
60
61     def really_delete_selected(self, request, queryset):
62         for obj in queryset:
63             obj.delete()
64
65         if queryset.count() == 1:
66             message_bit = "1 link entry was"
67         else:
68             message_bit = "%s link entries were" % queryset.count()
69         self.message_user(request, "%s successfully deleted." % message_bit)
70     really_delete_selected.short_description = "Delete selected entries"
71
72 admin.site.register(Results, ResultsAdmin)
73 admin.site.register(Keys, KeysAdmin)
74 admin.site.register(HeatmapLink, HeatmapsLinkAdmin)

```

**File:** ./health\_cloud\_back\_end/app/models.py

```

1 # -*- coding: utf-8 -*-
2 from __future__ import unicode_literals
3
4 from django.db import models
5 from django.db.models.deletion import CASCADE
6 from django.db.models.fields import CharField
7 from django.utils.crypto import get_random_string
8 from django.core.validators import MaxValueValidator, MinValueValidator
9 from django.db.models.signals import pre_delete
10 from django.dispatch import receiver
11 from ..settings import API_URL
12 import os
13
14 class Keys(models.Model):
15     key = models.CharField(('key'), max_length=20, blank=False)
16     counter = models.IntegerField(
17         default=0,
18         validators=[MaxValueValidator(10), MinValueValidator(0)]
19     )
20     created_at = models.DateTimeField(auto_now_add=True)
21
22     def save(self, *args, **kwargs):
23         if not self.pk:
24             self.key = get_random_string(length=20, allowed_chars='ABCDEFGHIIJKLMNOPQRSTUVWXYZ0123456789')
25             super(Keys, self).save(*args, **kwargs)
26
27     class Meta:
28         verbose_name = 'Key'
29         verbose_name_plural = 'Keys'
30
31     def __str__(self):
32         return self.key
33
34 class HeatmapLink(models.Model):
35     pathology = models.CharField(max_length=50, blank=False)
36     link = models.CharField(max_length=100, blank=False)
37
38     def delete(self, *args, **kwargs):
39         img_path = self.link.split(API_URL)[1]
40         if os.path.exists(img_path):
41             os.remove(img_path)
42         super(HeatmapLink, self).delete(*args, **kwargs)
43
44     class Meta:
45         verbose_name = 'HeatmapLink'
46         verbose_name_plural = 'HeatmapLinks'
47
48     def __str__(self):
49         return self.link
50
51 class Results(models.Model):
52     key = models.ForeignKey(Keys, on_delete=models.CASCADE)
53     originalImage = models.ImageField(upload_to='images')
54     heatmaps_links = models.ManyToManyField(HeatmapLink)
55     created_at = models.DateTimeField(auto_now_add=True)
56
57     class Meta:
58         verbose_name = 'Result'
59         verbose_name_plural = 'Results'
60
61     def __str__(self):
62         return str(self.key)
63
64 @receiver(pre_delete, sender=Results)
65 def delete_result(sender, instance, signal, *args, **kwargs):
66     instance.key.counter = instance.key.counter - 1
67     instance.key.save()
68     instance.originalImage.delete(save=False)
69
70     for heatmap_link in instance.heatmaps_links.all():
71         heatmap_link.delete()

```

#### File: ./health\_cloud\_back\_end/app/tests.py

```

1 # -*- coding: utf-8 -*-
2 from __future__ import unicode_literals
3
4 from django.test import TestCase
5
6 # Create your tests here.

```

#### File: ./health\_cloud\_back\_end/app/serializers.py

```

1 from rest_framework import serializers
2 from .models import Results
3
4 class ResultSerializer(serializers.HyperlinkedModelSerializer):
5     class Meta:
6         model = Results
7         fields = ('key', 'originalImage')

```

#### File: ./health\_cloud\_back\_end/app/views.py

```

1 from rest_framework import viewsets, mixins
2 from rest_framework.api_key.permissions import HasAPIKey
3 from django.http import HttpResponse
4 import json
5 from .models import Keys, Results
6 from .serializers import ResultSerializer
7 from .Almodel.model import get_results
8
9 class InferenceView(mixins.CreateModelMixin, viewsets.GenericViewSet):
10     queryset = Keys.objects.all().order_by('id')
11     serializer_class = ResultSerializer
12     permission_classes = [HasAPIKey]
13     filter_fields = ('_all_',)
14
15     def makeInference(self, request):
16         key = request.POST.dict()[key]
17
18         keyObj = Keys.objects.filter(key = key)
19
20         if keyObj:
21             if keyObj[0].counter < 10:
22                 originalImage = request.FILES.dict()[originalImage]
23
24                 ResultObj = Results.objects.create(key = keyObj[0], originalImage = originalImage)
25                 keyObj[0].counter = keyObj[0].counter + 1
26                 keyObj[0].save()
27
28                 try:
29                     results = get_results(ResultObj.originalImage.path)
30
31                     for result in results:
32                         print("Creating result for "+result['pathology'])
33                         ResultObj.heatmaps_links.create(pathology = result['pathology'], link=result['heatmap_link'])
34                         ResultObj.save()
35
36                 return HttpResponse(content=json.dumps(results, ensure_ascii=False).encode('utf8'), status=200)
37             except Exception as e:
38                 print(str(e))
39
40                 ResultObj.delete()
41
42                 return HttpResponse(content=json.dumps({'error':"Falha executando o modelo"}, ensure_ascii=False).encode('utf8'), status=400)
43         else:
44             return HttpResponse(content=json.dumps({'error':"Essa chave de acesso foi utilizada 10 vezes"}, ensure_ascii=False).encode('utf8'), status=400)
45     else:
46         return HttpResponse(content=json.dumps({'error':"Chave de acesso inválida"}, ensure_ascii=False).encode('utf8'), status=400)

```

**File: ./health\_cloud\_back\_end/app/Almodel/pretrained\_model.h5**

[binary]

**File: ./health\_cloud\_back\_end/app/Almodel/model.py**

```

1 import numpy as np
2 import cv2
3 from ...settings import BASE_DIR, API_URL
4 from django.utils.crypto import get_random_string
5
6 from tensorflow.keras.applications.densenet import DenseNet121
7 from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
8 from tensorflow.keras.models import Model
9 from tensorflow.keras.preprocessing import image
10 from tensorflow.keras import backend as K
11 from tensorflow.keras import models
12 from tensorflow import keras
13 import tensorflow as tf
14 from tensorflow.python.framework.ops import disable_eager_execution, enable_eager_execution
15
16 labels = [
17     'Cardiomegalia',
18     'Enfisema',
19     'Efuso',
20     'Hrnia',
21     'Infiltrao',
22     'Massa',
23     'Ndulo',
24     'Atelectasia',
25     'Pneumotrax',
26     'Espessamento Pleural',
27     'Pneumonia',
28     'Fibrose',
29     'Edema',
30     'Consolidao'
31 ]
32
33 def get_mean_std_per_batch(image_path, H=320, W=320):
34     sample_data = []
35     for idx in range(0, 100):
36         # path = image_dir + img
37         sample_data.append(np.array(image.load_img(image_path, target_size=(H, W))))
38
39     mean = np.mean(sample_data, 0)
40     std = np.std(sample_data, 0)
41     return mean, std
42
43 def load_image(img_path, preprocess=True, H=320, W=320):
44     """Load and preprocess image."""
45     mean, std = get_mean_std_per_batch(img_path, H=H, W=W)
46
47     x = image.load_img(img_path, target_size=(H, W))
48     if preprocess:
49         x -= mean
50         x /= std
51         x = np.expand_dims(x, axis=0)
52     return x
53
54 def generate_grad_cam(image_path, input_model, image, cls, layer_name, H=320, W=320):
55     """GradCAM method for visualizing input saliency."""
56     with tf.GradientTape() as gtape:
57         conv_layer = input_model.get_layer(layer_name)
58         heatmap_model = models.Model([input_model.inputs], [conv_layer.output, input_model.output])
59         conv_output, predictions = heatmap_model(image)
60         loss = predictions[0, cls]
61         grads = gtape.gradient(loss, conv_output)[0, :, :, :]
62         pooled_grads = np.mean(grads, axis=(0, 1))
63
64         cam = np.dot(conv_output[0, :, :], pooled_grads)
65
66     originalImage = cv2.imread(image_path)
67
68     cam = cv2.resize(cam, (originalImage.shape[1], originalImage.shape[0]), cv2.INTER_LINEAR)
69     cam = np.maximum(cam, 0)
70     cam = cam / cam.max()
71
72     resultImageName = "heatmap" + str(cls) + get_random_string(length=20, allowed_chars='ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789') + ".png"
73
74     cv2.imwrite('./images/' + resultImageName, 255 * cam)
75     heatmapImage = cv2.imread('./images/' + resultImageName)
76     heatmapImage = cv2.applyColorMap(heatmapImage, cv2.COLORMAP_JET)
77     result = cv2.addWeighted(originalImage, 0.7, heatmapImage, 0.3, 0)
78     cv2.imwrite('./images/' + resultImageName, result)
79
80     result_link = API_URL + "images/" + resultImageName
81
82     return result_link
83
84 def get_results(image_path):
85     base_model = DenseNet121(weights='imagenet', include_top=False)
86     x = base_model.output
87     x = GlobalAveragePooling2D()(x)
88     predictions = Dense(len(labels), activation="sigmoid")(x)
89     model = Model(inputs=base_model.input, outputs=predictions)
90     model.load_weights(BASE_DIR + "/health_cloud_back_end/app/Almodel/pretrained_model.h5")
91
92     preprocessed_input = load_image(image_path)
93     predictions = model.predict(preprocessed_input)[0]
94
95     results = []
96     for label in labels:
97         print("Running model for " + label)
98         cam = generate_grad_cam(image_path, model, preprocessed_input, labels.index(label), 'bn')
99         results.append(
100             {"heatmap_link": cam, "prediction": str(predictions[labels.index(label)]), "pathology": label}
101         )
102
103     return results

```

**File:** ./health\_cloud\_back\_end/app/migrations/0005\_results\_originalimage.py

```

1 # Generated by Django 3.1 on 2021-11-11 18:16
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0004_auto_20211111_1511'),
10    ]
11
12     operations = [
13         migrations.AddField(
14             model_name='results',
15             name='originalImage',
16             field=models.ImageField(default=0, upload_to='images'),
17             preserve_default=False,
18         ),
19    ]

```

**File: ./health\_cloud\_back\_end/app/migrations/0004\_auto\_20211111\_1511.py**

```

1 # Generated by Django 3.1 on 2021-11-11 18:11
2
3 import django.core.validators
4 from django.db import migrations, models
5 import django.db.models.deletion
6
7
8 class Migration(migrations.Migration):
9
10     dependencies = [
11         ('app', '0003_auto_20211111_0942'),
12    ]
13
14     operations = [
15         migrations.RemoveField(
16             model_name='keys',
17             name='results',
18         ),
19         migrations.AddField(
20             model_name='keys',
21             name='counter',
22             field=models.IntegerField(default=0, validators=[django.core.validators.MaxValueValidator(10), django.core.validators.MinValueValidator(0)]),
23         ),
24         migrations.AddField(
25             model_name='results',
26             name='key',
27             field=models.ForeignKey(default=0, on_delete=django.db.models.deletion.CASCADE, to='app.keys'),
28             preserve_default=False,
29         ),
30    ]

```

**File: ./health\_cloud\_back\_end/app/migrations/0011\_auto\_20211130\_2111.py**

```

1 # Generated by Django 3.1 on 2021-12-01 00:11
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0010_results_heatmaps'),
10    ]
11
12     operations = [
13         migrations.RemoveField(
14             model_name='results',
15             name='heatmaps',
16         ),
17         migrations.AddField(
18             model_name='results',
19             name='heatmaps_links',
20             field=models.CharField(blank=True, max_length=100),
21             preserve_default=False,
22         ),
23         migrations.DeleteModel(
24             name='Heatmap',
25         ),
26    ]

```

**File: ./health\_cloud\_back\_end/app/migrations/0012\_auto\_20211130\_2124.py**

```

1 # Generated by Django 3.1 on 2021-12-01 00:24
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0011_auto_20211130_2111'),
10    ]
11
12    operations = [
13        migrations.CreateModel(
14            name='HeatmapLink',
15            fields=[
16                ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
17                ('link', models.CharField(blank=True, max_length=100)),
18            ],
19            options={
20                'verbose_name': 'HeatmapLink',
21                'verbose_name_plural': 'HeatmapLinks',
22            },
23        ),
24        migrations.RemoveField(
25            model_name='results',
26            name='heatmaps_links',
27        ),
28        migrations.AddField(
29            model_name='results',
30            name='heatmaps_links',
31            field=models.ManyToManyField(to='app.HeatmapLink'),
32        ),
33    ]

```

**File: ./health\_cloud\_back\_end/app/migrations/0017\_remove\_heatmaplink\_result.py**

```

1 # Generated by Django 3.1 on 2021-12-01 01:59
2
3 from django.db import migrations
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0016_auto_20211130_2252'),
10    ]
11
12    operations = [
13        migrations.RemoveField(
14            model_name='heatmaplink',
15            name='result',
16        ),
17    ]

```

**File: ./health\_cloud\_back\_end/app/migrations/0016\_auto\_20211130\_2252.py**

```

1 # Generated by Django 3.1 on 2021-12-01 01:52
2
3 from django.db import migrations, models
4 import django.db.models.deletion
5
6
7 class Migration(migrations.Migration):
8
9     dependencies = [
10         ('app', '0015_heatmaplink_result'),
11    ]
12
13    operations = [
14        migrations.AlterField(
15            model_name='heatmaplink',
16            name='result',
17            field=models.ForeignKey(on_delete=django.db.models.deletion.CASCADE, to='app.results'),
18        ),
19    ]

```

**File: ./health\_cloud\_back\_end/app/migrations/0013\_auto\_20211130\_2240.py**

```

1 # Generated by Django 3.1 on 2021-12-01 01:40
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0012_auto_20211130_2124'),
10    ]
11
12    operations = [
13        migrations.AlterField(
14            model_name='heatmaplink',
15            name='link',
16            field=models.CharField(max_length=100),
17        ),
18    ]

```

**File: ./health\_cloud\_back\_end/app/migrations/0014\_heatmaplink\_pathology.py**



```

1 # Generated by Django 3.1 on 2021-12-01 01:41
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0013_auto_20211130_2240'),
10    ]
11
12    operations = [
13        migrations.AddField(
14            model_name='heatmaplink',
15            name='pathology',
16            field=models.CharField(default='memes', max_length=50),
17            preserve_default=False,
18        ),
19    ]

```

#### File: ./health\_cloud\_back\_end/app/migrations/0001\_initial.py

```

1 # Generated by Django 3.1 on 2021-11-11 12:21
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     initial = True
9
10    dependencies = [
11    ]
12
13    operations = [
14        migrations.CreateModel(
15            name='Results',
16            fields=[
17                ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
18                ('link', models.CharField(max_length=500, verbose_name='link')),
19            ],
20            options={
21                'verbose_name': 'Result',
22                'verbose_name_plural': 'Results',
23            },
24        ),
25    ]

```

#### File: ./health\_cloud\_back\_end/app/migrations/0009\_auto\_20211130\_1039.py

```

1 # Generated by Django 3.1 on 2021-11-30 13:39
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0008_result'),
10    ]
11
12    operations = [
13        migrations.CreateModel(
14            name='Heatmap',
15            fields=[
16                ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
17                ('pathology', models.CharField(max_length=50, verbose_name='pathology')),
18                ('image', models.ImageField(upload_to='images')),
19                ('created_at', models.DateTimeField(auto_now_add=True)),
20            ],
21            options={
22                'verbose_name': 'Heatmap',
23                'verbose_name_plural': 'Heatmaps',
24            },
25        ),
26        migrations.DeleteModel(
27            name='Result',
28        ),
29    ]

```

#### File: ./health\_cloud\_back\_end/app/migrations/0010\_results\_heatmaps.py

```

1 # Generated by Django 3.1 on 2021-11-30 14:11
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0009_auto_20211130_1039'),
10    ]
11
12    operations = [
13        migrations.AddField(
14            model_name='results',
15            name='heatmaps',
16            field=models.ManyToManyField(to='app.Heatmap'),
17        ),
18    ]

```

#### File: ./health\_cloud\_back\_end/app/migrations/0008\_result.py

```

1 # Generated by Django 3.1 on 2021-11-30 13:34
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0007_remove_results_link'),
10    ]
11
12     operations = [
13         migrations.CreateModel(
14             name='Result',
15             fields=[
16                 ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
17                 ('pathology', models.CharField(max_length=50, verbose_name='pathology')),
18                 ('image', models.ImageField(upload_to='images')),
19             ],
20             options={
21                 'verbose_name': 'Result',
22                 'verbose_name_plural': 'Results',
23             },
24         ),
25    ]

```

**File: ./health\_cloud\_back\_end/app/migrations/0007\_remove\_results\_link.py**

```

1 # Generated by Django 3.1 on 2021-11-30 13:16
2
3 from django.db import migrations
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0006_auto_20211130_0927'),
10    ]
11
12     operations = [
13         migrations.RemoveField(
14             model_name='results',
15             name='link',
16         ),
17    ]

```

**File: ./health\_cloud\_back\_end/app/migrations/0015\_heatmaplink\_result.py**

```

1 # Generated by Django 3.1 on 2021-12-01 01:51
2
3 from django.db import migrations, models
4 import django.db.models.deletion
5
6
7 class Migration(migrations.Migration):
8
9     dependencies = [
10         ('app', '0014_heatmaplink_pathology'),
11    ]
12
13     operations = [
14         migrations.AddField(
15             model_name='heatmaplink',
16             name='result',
17             field=models.ForeignKey(null=True, on_delete=django.db.models.deletion.CASCADE, to='app.results'),
18         ),
19    ]

```

**File: ./health\_cloud\_back\_end/app/migrations/0003\_auto\_20211111\_0942.py**

```

1 # Generated by Django 3.1 on 2021-11-11 12:42
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0002_keys'),
10    ]
11
12     operations = [
13         migrations.AlterField(
14             model_name='keys',
15             name='results',
16             field=models.ManyToManyField(blank=True, to='app.Results'),
17         ),
18    ]

```

**File: ./health\_cloud\_back\_end/app/migrations/0002\_keys.py**

```

1 # Generated by Django 3.1 on 2021-11-11 12:33
2
3 from django.db import migrations, models
4
5
6 class Migration(migrations.Migration):
7
8     dependencies = [
9         ('app', '0001_initial'),
10    ]
11
12    operations = [
13        migrations.CreateModel(
14            name='Keys',
15            fields=[
16                ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
17                ('key', models.CharField(max_length=20, verbose_name='key')),
18                ('results', models.ManyToManyField(to='app.Results')),
19            ],
20            options={
21                'verbose_name': 'Key',
22                'verbose_name_plural': 'Keys',
23            },
24        ),
25    ]

```

#### File: ./health\_cloud\_back\_end/app/migrations/0006\_auto\_20211130\_0927.py

```

1 # Generated by Django 3.1 on 2021-11-30 12:27
2
3 from django.db import migrations, models
4 import django.utils.timezone
5
6
7 class Migration(migrations.Migration):
8
9     dependencies = [
10         ('app', '0005_results_originalimage'),
11    ]
12
13    operations = [
14        migrations.AddField(
15            model_name='keys',
16            name='created_at',
17            field=models.DateTimeField(auto_now_add=True, default=django.utils.timezone.now),
18            preserve_default=False,
19        ),
20        migrations.AddField(
21            model_name='results',
22            name='created_at',
23            field=models.DateTimeField(auto_now_add=True, default=django.utils.timezone.now),
24            preserve_default=False,
25        ),
26    ]

```

#### File: ./health\_cloud\_back\_end/app/migrations/\_\_init\_\_.py

```
1 |binary|
```

#### File: ./health\_cloud\_back\_end/app/\_\_init\_\_.py

```
1 |binary|
```

#### File: ./health\_cloud\_back\_end/app/apps.py

```

1 # -*- coding: utf-8 -*-
2 from __future__ import unicode_literals
3
4 from django.apps import AppConfig
5
6
7 class AuthConfig(AppConfig):
8     name = 'health_cloud_back_end.app'

```

#### File: ./health\_cloud\_back\_end/settings.py

```

1 """
2 Django settings for health_cloud_back_end project.
3
4 Generated by 'django-admin startproject' using Django 1.11.29.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/1.11/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/1.11/ref/settings/
11 """
12
13 import os
14 import environ
15
16 env = environ.Env()
17 environ.Env.read_env()
18
19 # Build paths inside the project like this: os.path.join(BASE_DIR, ...)
20 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
21
22 # API_URL = "http://localhost:8000/"
23 API_URL = "http://ec2-3-140-18-246.us-east-2.compute.amazonaws.com/"
24
25
26 # Quick-start development settings - unsuitable for production

```

```

27 # See https://docs.djangoproject.com/en/1.11/howto/deployment/checklist/
28
29 # SECURITY WARNING: keep the secret key used in production secret!
30 SECRET_KEY = env('SECRET_KEY')
31
32 # SECURITY WARNING: don't run with debug turned on in production!
33 DEBUG = True
34
35 ALLOWED_HOSTS = ["ec2-3-140-18-246.us-east-2.compute.amazonaws.com"]
36
37 CORS_ORIGIN_ALLOW_ALL = True
38 CORS_ORIGIN_WHITELIST = (
39     'http://localhost',
40 )
41
42 # Application definition
43
44 INSTALLED_APPS = [
45     'django.contrib.admin',
46     'django.contrib.auth',
47     'django.contrib.contenttypes',
48     'django.contrib.sessions',
49     'django.contrib.messages',
50     'django.contrib.staticfiles',
51     'rest_framework',
52     'health_cloud_back_end.app',
53     'rest_framework_api_key',
54     'corsheaders'
55 ]
56
57 MIDDLEWARE = [
58     'django.middleware.security.SecurityMiddleware',
59     'django.contrib.sessions.middleware.SessionMiddleware',
60     'django.middleware.common.CommonMiddleware',
61     'django.middleware.csrf.CsrfViewMiddleware',
62     'django.contrib.auth.middleware.AuthenticationMiddleware',
63     'django.contrib.messages.middleware.MessageMiddleware',
64     'django.middleware.clickjacking.XFrameOptionsMiddleware',
65     'corsheaders.middleware.CorsMiddleware',
66 ]
67
68 ROOT_URLCONF = 'health_cloud_back_end.urls'
69
70 TEMPLATES = [
71     {
72         'BACKEND': 'django.template.backends.django.DjangoTemplates',
73         'DIRS': [],
74         'APP_DIRS': True,
75         'OPTIONS': {
76             'context_processors': [
77                 'django.template.context_processors.debug',
78                 'django.template.context_processors.request',
79                 'django.contrib.auth.context_processors.auth',
80                 'django.contrib.messages.context_processors.messages',
81             ],
82         },
83     },
84 ]
85
86 WSGI_APPLICATION = 'health_cloud_back_end.wsgi.application'
87
88
89 # Database
90 # https://docs.djangoproject.com/en/1.11/ref/settings/#databases
91
92 # DATABASES = {
93 #     'default': {
94 #         'ENGINE': 'django.db.backends.sqlite3',
95 #         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
96 #     }
97 # }
98
99 DATABASES = {
100     'default': {
101         'ENGINE': 'django.db.backends.postgresql_psycopg2',
102         'NAME': env('DATABASE_NAME'),
103         'USER': env('DATABASE_USER'),
104         'PASSWORD': env('DATABASE_PASS'),
105         'HOST': 'localhost',
106         'PORT': '',
107     }
108 }
109
110
111 # Password validation
112 # https://docs.djangoproject.com/en/1.11/ref/settings/#auth-password-validators
113
114 AUTH_PASSWORD_VALIDATORS = [
115     {
116         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
117     },
118     {
119         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
120     },
121     {
122         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
123     },
124     {
125         'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
126     },
127 ]
128
129 REST_FRAMEWORK = {
130     'DEFAULT_PAGINATION_CLASS': 'rest_framework.pagination.PageNumberPagination',
131     'PAGE_SIZE': 10,
132     'DEFAULT_PERMISSION_CLASSES': [
133         'rest_framework_api_key.permissions.HasAPIKey',
134     ]
135 }
136
137
138 # Internationalization
139 # https://docs.djangoproject.com/en/1.11/topics/i18n/
140
141 LANGUAGE_CODE = 'en-us'

```

```

142
143 TIME_ZONE = 'UTC'
144
145 USE_I18N = True
146
147 USE_L10N = True
148
149 USE_TZ = True
150
151
152 # Static files (CSS, JavaScript, Images)
153 # https://docs.djangoproject.com/en/1.11/howto/static-files/
154
155 STATIC_URL = '/static/'
156 STATIC_ROOT = "/var/www/health-cloud-back-end/static/"

```

#### File: ./health\_cloud\_back\_end/\_init\_\_.py

```

1 |binary|

```

#### File: ./health\_cloud\_back\_end/wsgi.py

```

1 """
2 WSGI config for health_cloud_back_end project.
3
4 It exposes the WSGI callable as a module-level variable named ``application``.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/1.11/howto/deployment/wsgi/
8 """
9
10 import os
11
12 from django.core.wsgi import get_wsgi_application
13
14 os.environ.setdefault("DJANGO_SETTINGS_MODULE", "health_cloud_back_end.settings")
15
16 application = get_wsgi_application()

```

#### File: ./static/rest\_framework/docs/css/base.css

```

1  h1 {
2      font-size: 45px;
3  }
4
5  .intro-code {
6      margin-top: 20px;
7  }
8
9  pre.highlight code * {
10     white-space: nowrap; /* this sets all children inside to nowrap */
11 }
12
13 pre.highlight {
14     overflow-x: auto; /* this sets the scrolling in x */
15 }
16
17 pre.highlight code {
18     white-space: pre; /* forces <code> to respect <pre> formatting */
19 }
20
21 .main-container {
22     padding-left: 30px;
23     padding-right: 30px;
24 }
25
26 .btn:focus {
27     outline: none;
28 }
29
30
31 .sidebar {
32     overflow: auto;
33     font-family: verdana, sans-serif;
34     font-size: 12px;
35     font-weight: 200;
36     background-color: #2e353d;
37     position: fixed;
38     top: 0px;
39     width: 225px;
40     height: 100%;
41     color: #FFF;
42 }
43
44 .sidebar .brand {
45     background-color: #23282e;
46     display: block;
47     text-align: center;
48     padding: 25px 0;
49     margin-top: 0;
50     margin-bottom: 0;
51 }
52
53 .sidebar .brand a {
54     color: #FFF;
55 }
56
57 .sidebar .brand a:hover,
58 .sidebar .brand a:active,
59 .sidebar .brand a:focus {
60     text-decoration: none;
61 }
62
63 .sidebar .toggle-btn {
64     display: none;
65 }
66

```

```

67 .sidebar .menu-list {
68   width: inherit;
69 }
70
71 .sidebar .menu-list ul,
72 .sidebar .menu-list li {
73   background: #2e353d;
74   list-style: none;
75   padding: 0px;
76   margin: 0px;
77   line-height: 35px;
78   cursor: pointer;
79 }
80
81 .sidebar .menu-list ul :not(collapsed) .arrow:before,
82 .sidebar .menu-list li :not(collapsed) .arrow:before {
83   font-family: FontAwesome;
84   content: "\f078";
85   display: inline-block;
86   padding-left: 10px;
87   padding-right: 10px;
88   vertical-align: middle;
89   float: right;
90 }
91
92 .sidebar .menu-list ul .active,
93 .sidebar .menu-list li .active {
94   border-left: 3px solid #d19b3d;
95   background-color: #4f5b69;
96 }
97
98 .sidebar .menu-list ul .sub-menu li.active,
99 .sidebar .menu-list li .sub-menu li.active {
100   color: #d19b3d;
101 }
102
103 .sidebar .menu-list ul .sub-menu li.active a,
104 .sidebar .menu-list li .sub-menu li.active a {
105   color: #d19b3d;
106 }
107
108 .sidebar .menu-list ul .sub-menu li,
109 .sidebar .menu-list li .sub-menu li {
110   background-color: #181c20;
111   border: none;
112   border-bottom: 1px solid #23282e;
113   margin-left: 0px;
114   line-height: 1.4;
115   padding-top: 10px;
116   padding-bottom: 10px;
117   padding-right: 10px;
118   padding-left: 25px;
119 }
120
121 .sidebar .menu-list ul .sub-menu li:hover,
122 .sidebar .menu-list li .sub-menu li:hover {
123   background-color: #020203;
124 }
125
126
127 .sidebar .menu-list ul .sub-menu li a,
128 .sidebar .menu-list li .sub-menu li a {
129   display: block;
130 }
131
132 .sidebar .menu-list ul .sub-menu li a:before,
133 .sidebar .menu-list li .sub-menu li a:before {
134   font-family: FontAwesome;
135   font-size: 14px;
136   font-weight: bold;
137   content: "\f105";
138   display: inline;
139   vertical-align: middle;
140   padding-left: 0;
141   padding-right: 7px;
142   margin-left: -12px;
143 }
144
145 .sidebar .menu-list li {
146   padding-left: 0px;
147   border-left: 3px solid #2e353d;
148   border-bottom: 1px solid #23282e;
149 }
150
151 .sidebar .menu-list li a {
152   text-decoration: none;
153   color: white;
154 }
155
156 .sidebar .menu-list li a i {
157   padding-left: 10px;
158   width: 20px;
159   padding-right: 20px;
160 }
161
162 .sidebar .menu-list li:hover {
163   border-left: 3px solid #d19b3d;
164   background-color: #4f5b69;
165   -webkit-transition: all 1s ease;
166   -moz-transition: all 1s ease;
167   -o-transition: all 1s ease;
168   -ms-transition: all 1s ease;
169   transition: all 1s ease;
170 }
171
172 .sidebar #menu-content {
173   padding-bottom: 70px;
174 }
175
176 body {
177   margin: 0px;
178   padding: 0px;
179 }
180
181 .coredocs-section-title {

```

```

182 margin-top: 20px;
183 padding-bottom: 10px;
184 border-bottom: 1px solid lightgrey;
185 }
186
187 coredocs-link-title a,
188 coredocs-section-title a {
189 display: none;
190 }
191
192 coredocs-link-title a,
193 coredocs-section-title a {
194 text-decoration: none;
195 }
196
197 coredocs-link-title:hover a,
198 coredocs-section-title:hover a {
199 display: inline;
200 font-size: 20px;
201 }
202
203 coredocs-section-title:last-child {
204 margin-top: 0;
205 }
206
207
208 /* @group Language Switcher */
209
210 .sidebar .menu-list.menu-list-bottom {
211 margin-bottom: 0;
212 position: fixed;
213 width: inherit;
214 bottom: 0;
215 left: 0;
216 right: 0;
217 border-top: 1px solid #23282e;
218 }
219
220 .sidebar .menu-list-bottom li span {
221 float: right;
222 margin-right: 20px;
223 color: #d19b3d;
224 }
225
226 /* @end Language Switcher */
227
228
229 /* @group Docs Content */
230
231 .docs-content .meta .label {
232 vertical-align: middle;
233 font-size: 14px;
234 font-weight: normal;
235 }
236
237 .docs-content .meta code {
238 vertical-align: middle;
239 padding: .2em .6em .3em;
240 font-size: 14px;
241 }
242
243 .docs-content .btn {
244 font-size: inherit;
245 }
246
247 .code-samples pre {
248 margin-top: 20px;
249 }
250
251 /* @end Docs Content */
252
253
254 @media (max-width: 767px) {
255 .main-container {
256 padding-left: 15px;
257 padding-right: 15px;
258 }
259
260 .sidebar {
261 position: relative;
262 width: 100%;
263 margin-bottom: 10px;
264 overflow: visible;
265 }
266
267 .sidebar .toggle-btn {
268 display: block;
269 cursor: pointer;
270 position: absolute;
271 right: 10px;
272 top: 10px;
273 z-index: 10 !important;
274 padding: 3px;
275 width: 40px;
276 text-align: center;
277 }
278
279 .sidebar .menu-list.menu-list-bottom {
280 position: static;
281 }
282
283 .sidebar .brand {
284 margin-top: 0;
285 margin-bottom: 0;
286
287 text-align: left !important;
288 font-size: 22px;
289 padding: 0;
290 padding-left: 20px;
291 line-height: 50px !important;
292 }
293
294
295 @media (min-width: 767px) {
296 .sidebar .menu-list .menu-content {

```

```

297     display: block;
298 }
299 #main {
300     width: calc(100% - 225px);
301     float: right;
302 }
303 }
304
305 @media (min-width: 992px) {
306     .modal-lg {
307         width: 980px;
308     }
309 }
310
311 .api-modal .modal-title .fa {
312     color: #93c54b;
313 }
314
315 .api-modal .modal-body .request-awaiting {
316     padding: 35px 10px;
317     color: #7f8177;
318     text-align: center;
319 }
320
321 .api-modal .modal-body .meta {
322     margin-bottom: 20px;
323 }
324
325 .api-modal .modal-body .meta .label {
326     vertical-align: middle;
327     font-size: 14px;
328     font-weight: normal;
329 }
330
331 .api-modal .modal-body .meta code {
332     vertical-align: middle;
333     padding: .2em .6em .3em;
334     font-size: 14px;
335 }
336
337 .api-modal .modal-content .toggle-view {
338     text-align: right;
339     float: right;
340 }
341
342 .api-modal .modal-content .response .well {
343     margin: 0;
344     max-height: 550px;
345 }
346
347 .highlight {
348     background-color: #f7f7f9
349 }
350
351 .checkbox label.control-label {
352     font-weight: bold;
353 }
354
355 @media (min-width: 768px) {
356     .navbar-nav.navbar-right:last-child {
357         margin-right: 0 !important;
358     }
359 }

```

#### File: ./static/rest\_framework/docs/css/jquery.json-view.min.css

```

1  .json-view{position: relative;}
2  .json-view .collapser {width: 20px; height: 18px; display: block; position: absolute; left: -1.7em; top: -.2em; z-index: 5; background-image: url(
3  .json-view .collapsed {-ms-transform: rotate(-90deg); -moz-transform: rotate(-90deg); -khtml-transform: rotate(-90deg); -webkit-transform: rotate(-90deg); -o-transform: rotate(-90deg); transform: rotate
4  .json-view .bl {display: block; padding-left: 20px; margin-left: -20px; position: relative;}
5  .json-view {font-family: monospace;}
6  .json-view ul {list-style-type: none; padding-left: 2em; border-left: 1px dotted; margin: .3em;}
7  .json-view ul li {position: relative;}
8  .json-view .comments, .json-view .dots {display: none; -moz-user-select: none; -ms-user-select: none; -khtml-user-select: none; -webkit-user-select: none; -o-user-select: none; user-select: none;}
9  .json-view .comments {padding-left: .8em; font-style: italic; color: #888;}
10 .json-view .bool, .json-view .null, .json-view .num, .json-view .undef {font-weight: 700; color: #1A01CC;}
11 .json-view .str {color: #800;}

```

#### File: ./static/rest\_framework/docs/css/highlight.css

```

1  /*
2  This is the GitHub theme for highlight.js
3
4  github.com style (c) Vasily Polovnyov <vast@whiteants.net>
5
6  */
7
8  .hljs {
9      display: block;
10     overflow-x: auto;
11     padding: 0.5em;
12     color: #333;
13     -webkit-text-size-adjust: none;
14 }
15
16 .hljs-comment,
17 .diff .hljs-header,
18 .hljs-javadoc {
19     color: #999;
20     font-style: italic;
21 }
22
23 .hljs-keyword,
24 .css .rule .hljs-keyword,
25 .hljs-winutils,
26 .nginx .hljs-title,

```



```

27 .hljs-subst,
28 .hljs-request,
29 .hljs-status {
30   color: #333;
31   font-weight: bold;
32 }
33
34 .hljs-number,
35 .hljs-hexcolor,
36 .ruby .hljs-constant {
37   color: #008080;
38 }
39
40 .hljs-string,
41 .hljs-tag .hljs-value,
42 .hljs-phpdoc,
43 .hljs-dartdoc,
44 .tex .hljs-formula {
45   color: #d14;
46 }
47
48 .hljs-title,
49 .hljs-id,
50 .scss .hljs-preprocessor {
51   color: #900;
52   font-weight: bold;
53 }
54
55 .hljs-list .hljs-keyword,
56 .hljs-subst {
57   font-weight: normal;
58 }
59
60 .hljs-class .hljs-title,
61 .hljs-type,
62 .vhdl .hljs-literal,
63 .tex .hljs-command {
64   color: #458;
65   font-weight: bold;
66 }
67
68 .hljs-tag,
69 .hljs-tag .hljs-title,
70 .hljs-rule .hljs-property,
71 .django .hljs-tag .hljs-keyword {
72   color: #000080;
73   font-weight: normal;
74 }
75
76 .hljs-attribute,
77 .hljs-variable,
78 .lisp .hljs-body,
79 .hljs-name {
80   color: #008080;
81 }
82
83 .hljs-regexp {
84   color: #009926;
85 }
86
87 .hljs-symbol,
88 .ruby .hljs-symbol .hljs-string,
89 .lisp .hljs-keyword,
90 .clojure .hljs-keyword,
91 .scheme .hljs-keyword,
92 .tex .hljs-special,
93 .hljs-prompt {
94   color: #990073;
95 }
96
97 .hljs-built_in {
98   color: #0086b3;
99 }
100
101 .hljs-preprocessor,
102 .hljs-pragma,
103 .hljs-pi,
104 .hljs-doctype,
105 .hljs-shebang,
106 .hljs-cdata {
107   color: #999;
108   font-weight: bold;
109 }
110
111 .hljs-deletion {
112   background: #ffd;
113 }
114
115 .hljs-addition {
116   background: #dfd;
117 }
118
119 .diff .hljs-change {
120   background: #0086b3;
121 }
122
123 .hljs-chunk {
124   color: #aaa;
125 }

```

**File:** `./static/rest_framework/docs/js/api.js`

```

1  var responseDisplay = 'data'
2  var coreapi = window.coreapi
3  var schema = window.schema
4
5  function normalizeKeys (arr) {
6    var _normarr = [];
7    for (var i = 0; i < arr.length; i++) {
8      _normarr = _normarr.concat(arr[i].split(' > '));
9    }
10   return _normarr;

```

```

11 }
12
13 function normalizeHTTPHeader (str) {
14   // Capitalize HTTP headers for display.
15   return (str.charAt(0) toUpperCase() + str.substring(1))
16     .replace(/-/g, function ($1) {
17       return $1 toUpperCase()
18     })
19     .replace(/(Www)/g, function ($1) {
20       return 'WWW'
21     })
22     .replace(/(Xss)/g, function ($1) {
23       return 'XSS'
24     })
25     .replace(/(Md5)/g, function ($1) {
26       return 'MD5'
27     })
28 }
29
30 function formEntries (form) {
31   // Polyfill for new FormData(form).entries()
32   var formData = new FormData(form)
33   if (formData.entries !== undefined) {
34     return Array.from(formData.entries())
35   }
36
37   var entries = []
38
39   for (var i = 0; i < form.elements.length; i++) {
40     var element = form.elements[i]
41
42     if (!element.name) {
43       continue
44     }
45
46     if (element.type === 'file') {
47       for (var j = 0; j < element.files.length; j++) {
48         entries.push([element.name, element.files[j]])
49       }
50     } else if (element.type === 'select-multiple' || element.type === 'select-one') {
51       for (var j = 0; j < element.selectedOptions.length; j++) {
52         entries.push([element.name, element.selectedOptions[j].value])
53       }
54     } else if (element.type === 'checkbox') {
55       if (element.checked) {
56         entries.push([element.name, element.value])
57       }
58     } else {
59       entries.push([element.name, element.value])
60     }
61   }
62
63   return entries
64 }
65
66 $(function () {
67   var $selectedAuthentication = $('#selected-authentication')
68   var $authControl = $('#auth-control')
69   var $authTokenModal = $('#auth_token_modal')
70   var $authBasicModal = $('#auth_basic_modal')
71   var $authSessionModal = $('#auth_session_modal')
72
73   // Language Control
74   $('#language-control li').click(function (event) {
75     event.preventDefault()
76     var $languageMenuItem = $(this).find('a')
77     var $languageControls = $(this).closest('ul').find('li')
78     var $languageControlLinks = $languageControls.find('a')
79     var language = $languageMenuItem.data('language')
80
81     $languageControlLinks.not('[data-language="' + language + '"]').parent().removeClass('active')
82     $languageControlLinks.filter('[data-language="' + language + '"]').parent().addClass('active')
83
84     $('#selected-language').text(language)
85
86     var $codeBlocks = $('pre.highlight')
87     $codeBlocks.not('[data-language="' + language + '"]').addClass('hide')
88     $codeBlocks.filter('[data-language="' + language + '"]').removeClass('hide')
89   })
90
91   // API Explorer
92   $('#form.api-interaction').submit(function (event) {
93     event.preventDefault()
94
95     var $form = $(this).closest('form')
96     var $requestMethod = $form.find('.request-method')
97     var $requestUrl = $form.find('.request-url')
98     var $toggleView = $form.closest('.modal-content').find('.toggle-view')
99     var $responseStatusCode = $form.find('.response-status-code')
100    var $meta = $form.find('.meta')
101    var $responseRawResponse = $form.find('.response-raw-response')
102    var $requestAwaiting = $form.find('.request-awaiting')
103    var $responseRaw = $form.find('.response-raw')
104    var $responseData = $form.find('.response-data')
105    var key = normalizeKeys($form.data('key'))
106    var params = {}
107    var entries = formEntries($form.get()[0])
108
109    for (var i = 0; i < entries.length; i++) {
110      var entry = entries[i]
111      var paramKey = entry[0]
112      var paramValue = entry[1]
113      var $elem = $form.find('[name="' + paramKey + '"]')
114      var dataType = $elem.data('type') || 'string'
115
116      if (dataType === 'integer' && paramValue) {
117        var value = parseInt(paramValue)
118        if (!isNaN(value)) {
119          params[paramKey] = value
120        }
121      } else if (dataType === 'number' && paramValue) {
122        var value = parseFloat(paramValue)
123        if (!isNaN(value)) {
124          params[paramKey] = value
125        }
126      }
127    }
128  })

```

```

126 } else if (dataType === 'boolean' && paramValue) {
127   var value = {
128     'true': true,
129     'false': false
130   }; paramValue.toLowerCase();
131   if (value !== undefined) {
132     params[paramKey] = value
133   }
134 } else if (dataType === 'array' && paramValue) {
135   try {
136     params[paramKey] = JSON.parse(paramValue)
137   } catch (err) {
138     // Ignore malformed JSON
139   }
140 } else if (dataType === 'object' && paramValue) {
141   try {
142     params[paramKey] = JSON.parse(paramValue)
143   } catch (err) {
144     // Ignore malformed JSON
145   }
146 } else if (dataType === 'string' && paramValue) {
147   params[paramKey] = paramValue
148 }
149 }
150
151 $form.find(':checkbox').each(function (index) {
152   // Handle unselected checkboxes
153   var name = $(this).attr('name')
154   if (!params.hasOwnProperty(name)) {
155     params[name] = false
156   }
157 })
158
159 function requestCallback (request) {
160   // Fill in the "GET /foo/" display.
161   var parser = document.createElement('a')
162   parser.href = request.url
163   var method = request.options.method
164   var path = parser.pathname + parser.hash + parser.search
165
166   $requestMethod.text(method)
167   $requestUrl.text(path)
168 }
169
170 function responseCallback (response, responseText) {
171   // Display the "Data"/"Raw" control.
172   $toggleView.removeClass('hide')
173
174   // Fill in the "200 OK" display.
175   $responseStatusCode.removeClass('label-success').removeClass('label-danger')
176   if (response.ok) {
177     $responseStatusCode.addClass('label-success')
178   } else {
179     $responseStatusCode.addClass('label-danger')
180   }
181   $responseStatusCode.text(response.status)
182   $meta.removeClass('hide')
183
184   // Fill in the Raw HTTP response display.
185   var panelText = 'HTTP/1.1' + response.status + ' ' + response.statusText + '\n'
186   response.headers.forEach(function (header, key) {
187     panelText += normalizeHTTPHeader(key) + ': ' + header + '\n'
188   })
189   if (responseText) {
190     panelText += '\n' + responseText
191   }
192   $responseRawResponse.text(panelText)
193 }
194
195 // Instantiate a client to make the outgoing request.
196 var options = {
197   requestCallback: requestCallback,
198   responseCallback: responseCallback
199 }
200
201 // Setup authentication options.
202 if (window.auth && window.auth.type === 'token') {
203   // Header authentication
204   options.auth = new coreapi.auth.TokenAuthentication({
205     scheme: window.auth.scheme,
206     token: window.auth.token
207   })
208 } else if (window.auth && window.auth.type === 'basic') {
209   // Basic authentication
210   options.auth = new coreapi.auth.BasicAuthentication({
211     username: window.auth.username,
212     password: window.auth.password
213   })
214 } else if (window.auth && window.auth.type === 'session') {
215   // Session authentication
216   options.auth = new coreapi.auth.SessionAuthentication({
217     csrfCookieName: 'csrftoken',
218     csrfHeaderName: 'X-CSRFToken'
219   })
220 }
221
222 var client = new coreapi.Client(options)
223 client.action(schema, key, params).then(function (data) {
224   var response = JSON.stringify(data, null, 2)
225   $requestAwaiting.addClass('hide')
226   $responseRaw.addClass('hide')
227   $responseData.addClass('hide').text('').jsonView(response)
228
229   if (responseDisplay === 'data') {
230     $responseData.removeClass('hide')
231   } else {
232     $responseRaw.removeClass('hide')
233   }
234 }) catch(function (error) {
235   var response = JSON.stringify(error.content, null, 2)
236   $requestAwaiting.addClass('hide')
237   $responseRaw.addClass('hide')
238   $responseData.addClass('hide').text('').jsonView(response)
239
240   if (responseDisplay === 'data') {

```

```

241     $responseData.removeClass('hide')
242   } else {
243     $responseRaw.removeClass('hide')
244   }
245 })
246 })
247
248 // 'Data'/'Raw' control
249 $('toggle-view button').click(function () {
250   var $modalContent = $(this).closest('.modal-content')
251   var $modalResponseRaw = $modalContent.find('.response-raw')
252   var $modalResponseData = $modalContent.find('.response-data')
253
254   responseDisplay = $(this).data('display-toggle')
255
256   $(this).removeClass('btn-default').addClass('btn-info').siblings().removeClass('btn-info')
257
258   if (responseDisplay === 'raw') {
259     $modalResponseRaw.removeClass('hide')
260     $modalResponseData.addClass('hide')
261   } else {
262     $modalResponseData.removeClass('hide')
263     $modalResponseRaw.addClass('hide')
264   }
265 })
266
267 // Authentication: none
268 $authControl.find("[data-auth='none']").click(function (event) {
269   event.preventDefault()
270   window.auth = null
271   $selectedAuthentication.text('none')
272   $authControl.find("[data-auth='']").closest('li').removeClass('active')
273   $authControl.find("[data-auth='none']").closest('li').addClass('active')
274 })
275
276 // Authentication: token
277 $('form.authentication-token-form').submit(function (event) {
278   event.preventDefault()
279   var $form = $(this).closest('form')
280   var scheme = $form.find('input#scheme').val()
281   var token = $form.find('input#token').val()
282   window.auth = {
283     'type': 'token',
284     'scheme': scheme,
285     'token': token
286   }
287   $selectedAuthentication.text('token')
288   $authControl.find("[data-auth='']").closest('li').removeClass('active')
289   $authControl.find("[data-auth='token']").closest('li').addClass('active')
290   $authTokenModal.modal('hide')
291 })
292
293 // Authentication: basic
294 $('form.authentication-basic-form').submit(function (event) {
295   event.preventDefault()
296   var $form = $(this).closest('form')
297   var username = $form.find('input#username').val()
298   var password = $form.find('input#password').val()
299   window.auth = {
300     'type': 'basic',
301     'username': username,
302     'password': password
303   }
304   $selectedAuthentication.text('basic')
305   $authControl.find("[data-auth='']").closest('li').removeClass('active')
306   $authControl.find("[data-auth='basic']").closest('li').addClass('active')
307   $authBasicModal.modal('hide')
308 })
309
310 // Authentication: session
311 $('form.authentication-session-form').submit(function (event) {
312   event.preventDefault()
313   window.auth = {
314     'type': 'session'
315   }
316   $selectedAuthentication.text('session')
317   $authControl.find("[data-auth='']").closest('li').removeClass('active')
318   $authControl.find("[data-auth='session']").closest('li').addClass('active')
319   $authSessionModal.modal('hide')
320 })
321 })

```

**File: ./static/rest\_framework/docs/js/jquery.json-view.min.js**

```

1 /**
2  * jquery.json-view - jQuery collapsible JSON plugin
3  * @version v1.0.0
4  * @link http://github.com/bazh/jquery.json-view
5  * @license MIT
6  */
7 function(e){if("use strict",var n=function(n){var a=e("<span />","class","collapser",on:{click:function(){var n=e(this);n.toggleClass("collapsed");var a=n.parent().children(".block")p=a.children("u

```

**File: ./static/rest\_framework/docs/js/highlight.pack.js**

```

1 |binary|

```

**File: ./static/rest\_framework/docs/img/favicon.ico**

```

|binary|

```

**File: ./static/rest\_framework/docs/img/grid.png**

```

|binary|

```

**File: ./static/rest\_framework/css/font-awesome-4.0.3.css**

```
1  /*!
2  * Font Awesome 4.0.3 by @davegandy - http://fontawesome.io - @fontawesome
3  * License - http://fontawesome.io/license (Font: SIL OFL 1.1, CSS: MIT License)
4  */
5  /* FONT PATH
6  * ..... */
7  @font-face {
8    font-family: 'FontAwesome';
9    src: url('../fonts/fontawesome-webfont.eot?v=4.0.3');
10   src: url('../fonts/fontawesome-webfont.eot?#iefix&v=4.0.3') format('embedded-opentype'), url('../fonts/fontawesome-webfont.woff?v=4.0.3') format('woff'), url('../fonts/fontawesome-webfont.
11   font-weight: normal;
12   font-style: normal;
13 }
14 .fa {
15   display: inline-block;
16   font-family: FontAwesome;
17   font-style: normal;
18   font-weight: normal;
19   line-height: 1;
20   -webkit-font-smoothing: antialiased;
21   -moz-osx-font-smoothing: grayscale;
22 }
23 /* makes the font 33% larger relative to the icon container */
24 .fa-lg {
25   font-size: 1.3333333333333333em;
26   line-height: 0.75em;
27   vertical-align: -15%;
28 }
29 .fa-2x {
30   font-size: 2em;
31 }
32 .fa-3x {
33   font-size: 3em;
34 }
35 .fa-4x {
36   font-size: 4em;
37 }
38 .fa-5x {
39   font-size: 5em;
40 }
41 .fa-fw {
42   width: 1.2857142857142858em;
43   text-align: center;
44 }
45 .fa-ul {
46   padding-left: 0;
47   margin-left: 2.142857142857143em;
48   list-style-type: none;
49 }
50 .fa-ul > li {
51   position: relative;
52 }
53 .fa-li {
54   position: absolute;
55   left: -2.142857142857143em;
56   width: 2.142857142857143em;
57   top: 0.14285714285714285em;
58   text-align: center;
59 }
60 .fa-li.fa-lg {
61   left: -1.8571428571428572em;
62 }
63 .fa-border {
64   padding: .2em .25em .15em;
65   border: solid 0.08em #eeeeee;
66   border-radius: .1em;
67 }
68 .pull-right {
69   float: right;
70 }
71 .pull-left {
72   float: left;
73 }
74 .fa.pull-left {
75   margin-right: .3em;
76 }
77 .fa.pull-right {
78   margin-left: .3em;
79 }
80 .fa-spin {
81   -webkit-animation: spin 2s infinite linear;
82   -moz-animation: spin 2s infinite linear;
83   -o-animation: spin 2s infinite linear;
84   animation: spin 2s infinite linear;
85 }
86 @-moz-keyframes spin {
87   0% {
88     -moz-transform: rotate(0deg);
89   }
90   100% {
91     -moz-transform: rotate(359deg);
92   }
93 }
94 @-webkit-keyframes spin {
95   0% {
96     -webkit-transform: rotate(0deg);
97   }
98   100% {
99     -webkit-transform: rotate(359deg);
100  }
101 }
102 @-o-keyframes spin {
103   0% {
104     -o-transform: rotate(0deg);
105   }
106   100% {
107     -o-transform: rotate(359deg);
108   }
109 }
```

```

110 @-ms-keyframes spin {
111   0% {
112     -ms-transform: rotate(0deg);
113   }
114   100% {
115     -ms-transform: rotate(359deg);
116   }
117 }
118 @keyframes spin {
119   0% {
120     transform: rotate(0deg);
121   }
122   100% {
123     transform: rotate(359deg);
124   }
125 }
126 .fa-rotate-90 {
127   filter: progid:DXImageTransform.Microsoft.BasicImage(rotation=1);
128   -webkit-transform: rotate(90deg);
129   -moz-transform: rotate(90deg);
130   -ms-transform: rotate(90deg);
131   -o-transform: rotate(90deg);
132   transform: rotate(90deg);
133 }
134 .fa-rotate-180 {
135   filter: progid:DXImageTransform.Microsoft.BasicImage(rotation=2);
136   -webkit-transform: rotate(180deg);
137   -moz-transform: rotate(180deg);
138   -ms-transform: rotate(180deg);
139   -o-transform: rotate(180deg);
140   transform: rotate(180deg);
141 }
142 .fa-rotate-270 {
143   filter: progid:DXImageTransform.Microsoft.BasicImage(rotation=3);
144   -webkit-transform: rotate(270deg);
145   -moz-transform: rotate(270deg);
146   -ms-transform: rotate(270deg);
147   -o-transform: rotate(270deg);
148   transform: rotate(270deg);
149 }
150 .fa-flip-horizontal {
151   filter: progid:DXImageTransform.Microsoft.BasicImage(rotation=0, mirror=1);
152   -webkit-transform: scale(-1, 1);
153   -moz-transform: scale(-1, 1);
154   -ms-transform: scale(-1, 1);
155   -o-transform: scale(-1, 1);
156   transform: scale(-1, 1);
157 }
158 .fa-flip-vertical {
159   filter: progid:DXImageTransform.Microsoft.BasicImage(rotation=2, mirror=1);
160   -webkit-transform: scale(1, -1);
161   -moz-transform: scale(1, -1);
162   -ms-transform: scale(1, -1);
163   -o-transform: scale(1, -1);
164   transform: scale(1, -1);
165 }
166 .fa-stack {
167   position: relative;
168   display: inline-block;
169   width: 2em;
170   height: 2em;
171   line-height: 2em;
172   vertical-align: middle;
173 }
174 .fa-stack-1x,
175 .fa-stack-2x {
176   position: absolute;
177   left: 0;
178   width: 100%;
179   text-align: center;
180 }
181 .fa-stack-1x {
182   line-height: inherit;
183 }
184 .fa-stack-2x {
185   font-size: 2em;
186 }
187 .fa-inverse {
188   color: #ffffff;
189 }
190 /* Font Awesome uses the Unicode Private Use Area (PUA) to ensure screen
191    readers do not read off random characters that represent icons */
192 .fa-glass:before {
193   content: "\f000";
194 }
195 .fa-music:before {
196   content: "\f001";
197 }
198 .fa-search:before {
199   content: "\f002";
200 }
201 .fa-envelope-o:before {
202   content: "\f003";
203 }
204 .fa-heart:before {
205   content: "\f004";
206 }
207 .fa-star:before {
208   content: "\f005";
209 }
210 .fa-star-o:before {
211   content: "\f006";
212 }
213 .fa-user:before {
214   content: "\f007";
215 }
216 .fa-film:before {
217   content: "\f008";
218 }
219 .fa-th-large:before {
220   content: "\f009";
221 }
222 .fa-th:before {
223   content: "\f00a";
224 }

```

```
225 .fa-th-list:before {
226   content: "\f00b";
227 }
228 .fa-check:before {
229   content: "\f00c";
230 }
231 .fa-times:before {
232   content: "\f00d";
233 }
234 .fa-search-plus:before {
235   content: "\f00e";
236 }
237 .fa-search-minus:before {
238   content: "\f010";
239 }
240 .fa-power-off:before {
241   content: "\f011";
242 }
243 .fa-signal:before {
244   content: "\f012";
245 }
246 .fa-gear:before
247 .fa-cog:before {
248   content: "\f013";
249 }
250 .fa-trash-o:before {
251   content: "\f014";
252 }
253 .fa-home:before {
254   content: "\f015";
255 }
256 .fa-file-o:before {
257   content: "\f016";
258 }
259 .fa-clock-o:before {
260   content: "\f017";
261 }
262 .fa-road:before {
263   content: "\f018";
264 }
265 .fa-download:before {
266   content: "\f019";
267 }
268 .fa-arrow-circle-o-down:before {
269   content: "\f01a";
270 }
271 .fa-arrow-circle-o-up:before {
272   content: "\f01b";
273 }
274 .fa-inbox:before {
275   content: "\f01c";
276 }
277 .fa-play-circle-o:before {
278   content: "\f01d";
279 }
280 .fa-rotate-right:before,
281 .fa-repeat:before {
282   content: "\f01e";
283 }
284 .fa-refresh:before {
285   content: "\f021";
286 }
287 .fa-list-alt:before {
288   content: "\f022";
289 }
290 .fa-lock:before {
291   content: "\f023";
292 }
293 .fa-flag:before {
294   content: "\f024";
295 }
296 .fa-headphones:before {
297   content: "\f025";
298 }
299 .fa-volume-off:before {
300   content: "\f026";
301 }
302 .fa-volume-down:before {
303   content: "\f027";
304 }
305 .fa-volume-up:before {
306   content: "\f028";
307 }
308 .fa-qrcode:before {
309   content: "\f029";
310 }
311 .fa-barcode:before {
312   content: "\f02a";
313 }
314 .fa-tag:before {
315   content: "\f02b";
316 }
317 .fa-tags:before {
318   content: "\f02c";
319 }
320 .fa-book:before {
321   content: "\f02d";
322 }
323 .fa-bookmark:before {
324   content: "\f02e";
325 }
326 .fa-print:before {
327   content: "\f02f";
328 }
329 .fa-camera:before {
330   content: "\f030";
331 }
332 .fa-font:before {
333   content: "\f031";
334 }
335 .fa-bold:before {
336   content: "\f032";
337 }
338 .fa-italic:before {
339   content: "\f033";
```

```
340 }
341 .fa-text-height:before {
342   content: "\f034";
343 }
344 .fa-text-width:before {
345   content: "\f035";
346 }
347 .fa-align-left:before {
348   content: "\f036";
349 }
350 .fa-align-center:before {
351   content: "\f037";
352 }
353 .fa-align-right:before {
354   content: "\f038";
355 }
356 .fa-align-justify:before {
357   content: "\f039";
358 }
359 .fa-list:before {
360   content: "\f03a";
361 }
362 .fa-dedent:before
363 .fa-outdent:before {
364   content: "\f03b";
365 }
366 .fa-indent:before {
367   content: "\f03c";
368 }
369 .fa-video-camera:before {
370   content: "\f03d";
371 }
372 .fa-picture-o:before {
373   content: "\f03e";
374 }
375 .fa-pencil:before {
376   content: "\f040";
377 }
378 .fa-map-marker:before {
379   content: "\f041";
380 }
381 .fa-adjust:before {
382   content: "\f042";
383 }
384 .fa-tint:before {
385   content: "\f043";
386 }
387 .fa-edit:before
388 .fa-pencil-square-o:before {
389   content: "\f044";
390 }
391 .fa-share-square-o:before {
392   content: "\f045";
393 }
394 .fa-check-square-o:before {
395   content: "\f046";
396 }
397 .fa-arrows:before {
398   content: "\f047";
399 }
400 .fa-step-backward:before {
401   content: "\f048";
402 }
403 .fa-fast-backward:before {
404   content: "\f049";
405 }
406 .fa-backward:before {
407   content: "\f04a";
408 }
409 .fa-play:before {
410   content: "\f04b";
411 }
412 .fa-pause:before {
413   content: "\f04c";
414 }
415 .fa-stop:before {
416   content: "\f04d";
417 }
418 .fa-forward:before {
419   content: "\f04e";
420 }
421 .fa-fast-forward:before {
422   content: "\f050";
423 }
424 .fa-step-forward:before {
425   content: "\f051";
426 }
427 .fa-eject:before {
428   content: "\f052";
429 }
430 .fa-chevron-left:before {
431   content: "\f053";
432 }
433 .fa-chevron-right:before {
434   content: "\f054";
435 }
436 .fa-plus-circle:before {
437   content: "\f055";
438 }
439 .fa-minus-circle:before {
440   content: "\f056";
441 }
442 .fa-times-circle:before {
443   content: "\f057";
444 }
445 .fa-check-circle:before {
446   content: "\f058";
447 }
448 .fa-question-circle:before {
449   content: "\f059";
450 }
451 .fa-info-circle:before {
452   content: "\f05a";
453 }
454 .fa-crosshairs:before {
```



```
455 content: "\f05b";
456 }
457 .fa-times-circle-o:before {
458 content: "\f05c";
459 }
460 .fa-check-circle-o:before {
461 content: "\f05d";
462 }
463 .fa-ban:before {
464 content: "\f05e";
465 }
466 .fa-arrow-left:before {
467 content: "\f060";
468 }
469 .fa-arrow-right:before {
470 content: "\f061";
471 }
472 .fa-arrow-up:before {
473 content: "\f062";
474 }
475 .fa-arrow-down:before {
476 content: "\f063";
477 }
478 .fa-mail-forward:before
479 .fa-share:before {
480 content: "\f064";
481 }
482 .fa-expand:before {
483 content: "\f065";
484 }
485 .fa-compress:before {
486 content: "\f066";
487 }
488 .fa-plus:before {
489 content: "\f067";
490 }
491 .fa-minus:before {
492 content: "\f068";
493 }
494 .fa-asterisk:before {
495 content: "\f069";
496 }
497 .fa-exclamation-circle:before {
498 content: "\f06a";
499 }
500 .fa-gift:before {
501 content: "\f06b";
502 }
503 .fa-leaf:before {
504 content: "\f06c";
505 }
506 .fa-fire:before {
507 content: "\f06d";
508 }
509 .fa-eye:before {
510 content: "\f06e";
511 }
512 .fa-eye-slash:before {
513 content: "\f070";
514 }
515 .fa-warning:before
516 .fa-exclamation-triangle:before {
517 content: "\f071";
518 }
519 .fa-plane:before {
520 content: "\f072";
521 }
522 .fa-calendar:before {
523 content: "\f073";
524 }
525 .fa-random:before {
526 content: "\f074";
527 }
528 .fa-comment:before {
529 content: "\f075";
530 }
531 .fa-magnet:before {
532 content: "\f076";
533 }
534 .fa-chevron-up:before {
535 content: "\f077";
536 }
537 .fa-chevron-down:before {
538 content: "\f078";
539 }
540 .fa-retweet:before {
541 content: "\f079";
542 }
543 .fa-shopping-cart:before {
544 content: "\f07a";
545 }
546 .fa-folder:before {
547 content: "\f07b";
548 }
549 .fa-folder-open:before {
550 content: "\f07c";
551 }
552 .fa-arrows-v:before {
553 content: "\f07d";
554 }
555 .fa-arrows-h:before {
556 content: "\f07e";
557 }
558 .fa-bar-chart-o:before {
559 content: "\f080";
560 }
561 .fa-twitter-square:before {
562 content: "\f081";
563 }
564 .fa-facebook-square:before {
565 content: "\f082";
566 }
567 .fa-camera-retro:before {
568 content: "\f083";
569 }
```

```
570 .fa-key:before {
571   content: "\f084";
572 }
573 .fa-gears:before {
574 .fa-cogs:before {
575   content: "\f085";
576 }
577 .fa-comments:before {
578   content: "\f086";
579 }
580 .fa-thumbs-o-up:before {
581   content: "\f087";
582 }
583 .fa-thumbs-o-down:before {
584   content: "\f088";
585 }
586 .fa-star-half:before {
587   content: "\f089";
588 }
589 .fa-heart-o:before {
590   content: "\f08a";
591 }
592 .fa-sign-out:before {
593   content: "\f08b";
594 }
595 .fa-linkedin-square:before {
596   content: "\f08c";
597 }
598 .fa-thumb-tack:before {
599   content: "\f08d";
600 }
601 .fa-external-link:before {
602   content: "\f08e";
603 }
604 .fa-sign-in:before {
605   content: "\f090";
606 }
607 .fa-trophy:before {
608   content: "\f091";
609 }
610 .fa-github-square:before {
611   content: "\f092";
612 }
613 .fa-upload:before {
614   content: "\f093";
615 }
616 .fa-lemon-o:before {
617   content: "\f094";
618 }
619 .fa-phone:before {
620   content: "\f095";
621 }
622 .fa-square-o:before {
623   content: "\f096";
624 }
625 .fa-bookmark-o:before {
626   content: "\f097";
627 }
628 .fa-phone-square:before {
629   content: "\f098";
630 }
631 .fa-twitter:before {
632   content: "\f099";
633 }
634 .fa-facebook:before {
635   content: "\f09a";
636 }
637 .fa-github:before {
638   content: "\f09b";
639 }
640 .fa-unlock:before {
641   content: "\f09c";
642 }
643 .fa-credit-card:before {
644   content: "\f09d";
645 }
646 .fa-rss:before {
647   content: "\f09e";
648 }
649 .fa-hdd-o:before {
650   content: "\f0a0";
651 }
652 .fa-bullhorn:before {
653   content: "\f0a1";
654 }
655 .fa-bell:before {
656   content: "\f0f3";
657 }
658 .fa-certificate:before {
659   content: "\f0a3";
660 }
661 .fa-hand-o-right:before {
662   content: "\f0a4";
663 }
664 .fa-hand-o-left:before {
665   content: "\f0a5";
666 }
667 .fa-hand-o-up:before {
668   content: "\f0a6";
669 }
670 .fa-hand-o-down:before {
671   content: "\f0a7";
672 }
673 .fa-arrow-circle-left:before {
674   content: "\f0a8";
675 }
676 .fa-arrow-circle-right:before {
677   content: "\f0a9";
678 }
679 .fa-arrow-circle-up:before {
680   content: "\f0aa";
681 }
682 .fa-arrow-circle-down:before {
683   content: "\f0ab";
684 }
```

```
685 .fa-globe:before {
686   content: "\f0ac";
687 }
688 .fa-wrench:before {
689   content: "\f0ad";
690 }
691 .fa-tasks:before {
692   content: "\f0ae";
693 }
694 .fa-filter:before {
695   content: "\f0b0";
696 }
697 .fa-briefcase:before {
698   content: "\f0b1";
699 }
700 .fa-arrows-alt:before {
701   content: "\f0b2";
702 }
703 .fa-group:before
704 .fa-users:before {
705   content: "\f0c0";
706 }
707 .fa-chain:before
708 .fa-link:before {
709   content: "\f0c1";
710 }
711 .fa-cloud:before {
712   content: "\f0c2";
713 }
714 .fa-flask:before {
715   content: "\f0c3";
716 }
717 .fa-cut:before
718 .fa-scissors:before {
719   content: "\f0c4";
720 }
721 .fa-copy:before
722 .fa-files-o:before {
723   content: "\f0c5";
724 }
725 .fa-paperclip:before {
726   content: "\f0c6";
727 }
728 .fa-save:before
729 .fa-floppy-o:before {
730   content: "\f0c7";
731 }
732 .fa-square:before {
733   content: "\f0c8";
734 }
735 .fa-bars:before {
736   content: "\f0c9";
737 }
738 .fa-list-ul:before {
739   content: "\f0ca";
740 }
741 .fa-list-ol:before {
742   content: "\f0cb";
743 }
744 .fa-strikethrough:before {
745   content: "\f0cc";
746 }
747 .fa-underline:before {
748   content: "\f0cd";
749 }
750 .fa-table:before {
751   content: "\f0ce";
752 }
753 .fa-magic:before {
754   content: "\f0d0";
755 }
756 .fa-truck:before {
757   content: "\f0d1";
758 }
759 .fa-pinterest:before {
760   content: "\f0d2";
761 }
762 .fa-pinterest-square:before {
763   content: "\f0d3";
764 }
765 .fa-google-plus-square:before {
766   content: "\f0d4";
767 }
768 .fa-google-plus:before {
769   content: "\f0d5";
770 }
771 .fa-money:before {
772   content: "\f0d6";
773 }
774 .fa-caret-down:before {
775   content: "\f0d7";
776 }
777 .fa-caret-up:before {
778   content: "\f0d8";
779 }
780 .fa-caret-left:before {
781   content: "\f0d9";
782 }
783 .fa-caret-right:before {
784   content: "\f0da";
785 }
786 .fa-columns:before {
787   content: "\f0db";
788 }
789 .fa-unsorted:before
790 .fa-sort:before {
791   content: "\f0dc";
792 }
793 .fa-sort-down:before
794 .fa-sort-asc:before {
795   content: "\f0dd";
796 }
797 .fa-sort-up:before
798 .fa-sort-desc:before {
799   content: "\f0de";
```

```
800 }
801 .fa-envelope:before {
802   content: "\f0e0";
803 }
804 .fa-linkedin:before {
805   content: "\f0e1";
806 }
807 .fa-rotate-left:before
808 .fa-undo:before {
809   content: "\f0e2";
810 }
811 .fa-legal:before
812 .fa-gavel:before {
813   content: "\f0e3";
814 }
815 .fa-dashboard:before
816 .fa-tachometer:before {
817   content: "\f0e4";
818 }
819 .fa-comment-o:before {
820   content: "\f0e5";
821 }
822 .fa-comments-o:before {
823   content: "\f0e6";
824 }
825 .fa-flash:before
826 .fa-bolt:before {
827   content: "\f0e7";
828 }
829 .fa-sitemap:before {
830   content: "\f0e8";
831 }
832 .fa-umbrella:before {
833   content: "\f0e9";
834 }
835 .fa-paste:before
836 .fa-clipboard:before {
837   content: "\f0ea";
838 }
839 .fa-lightbulb-o:before {
840   content: "\f0eb";
841 }
842 .fa-exchange:before {
843   content: "\f0ec";
844 }
845 .fa-cloud-download:before {
846   content: "\f0ed";
847 }
848 .fa-cloud-upload:before {
849   content: "\f0ee";
850 }
851 .fa-user-md:before {
852   content: "\f0f0";
853 }
854 .fa-stethoscope:before {
855   content: "\f0f1";
856 }
857 .fa-suitcase:before {
858   content: "\f0f2";
859 }
860 .fa-bell-o:before {
861   content: "\f0a2";
862 }
863 .fa-coffee:before {
864   content: "\f0f4";
865 }
866 .fa-cutlery:before {
867   content: "\f0f5";
868 }
869 .fa-file-text-o:before {
870   content: "\f0f6";
871 }
872 .fa-building-o:before {
873   content: "\f0f7";
874 }
875 .fa-hospital-o:before {
876   content: "\f0f8";
877 }
878 .fa-ambulance:before {
879   content: "\f0f9";
880 }
881 .fa-medkit:before {
882   content: "\f0fa";
883 }
884 .fa-fighter-jet:before {
885   content: "\f0fb";
886 }
887 .fa-beer:before {
888   content: "\f0fc";
889 }
890 .fa-h-square:before {
891   content: "\f0fd";
892 }
893 .fa-plus-square:before {
894   content: "\f0fe";
895 }
896 .fa-angle-double-left:before {
897   content: "\f100";
898 }
899 .fa-angle-double-right:before {
900   content: "\f101";
901 }
902 .fa-angle-double-up:before {
903   content: "\f102";
904 }
905 .fa-angle-double-down:before {
906   content: "\f103";
907 }
908 .fa-angle-left:before {
909   content: "\f104";
910 }
911 .fa-angle-right:before {
912   content: "\f105";
913 }
914 .fa-angle-up:before {
```

```
915 content: "\f106";
916 }
917 .fa-angle-down:before {
918 content: "\f107";
919 }
920 .fa-desktop:before {
921 content: "\f108";
922 }
923 .fa-laptop:before {
924 content: "\f109";
925 }
926 .fa-tablet:before {
927 content: "\f10a";
928 }
929 .fa-mobile-phone:before,
930 .fa-mobile:before {
931 content: "\f10b";
932 }
933 .fa-circle-o:before {
934 content: "\f10c";
935 }
936 .fa-quote-left:before {
937 content: "\f10d";
938 }
939 .fa-quote-right:before {
940 content: "\f10e";
941 }
942 .fa-spinner:before {
943 content: "\f110";
944 }
945 .fa-circle:before {
946 content: "\f111";
947 }
948 .fa-mail-reply:before,
949 .fa-reply:before {
950 content: "\f112";
951 }
952 .fa-github-alt:before {
953 content: "\f113";
954 }
955 .fa-folder-o:before {
956 content: "\f114";
957 }
958 .fa-folder-open-o:before {
959 content: "\f115";
960 }
961 .fa-smile-o:before {
962 content: "\f118";
963 }
964 .fa-frown-o:before {
965 content: "\f119";
966 }
967 .fa-meh-o:before {
968 content: "\f11a";
969 }
970 .fa-gamepad:before {
971 content: "\f11b";
972 }
973 .fa-keyboard-o:before {
974 content: "\f11c";
975 }
976 .fa-flag-o:before {
977 content: "\f11d";
978 }
979 .fa-flag-checkered:before {
980 content: "\f11e";
981 }
982 .fa-terminal:before {
983 content: "\f120";
984 }
985 .fa-code:before {
986 content: "\f121";
987 }
988 .fa-reply-all:before {
989 content: "\f122";
990 }
991 .fa-mail-reply-all:before {
992 content: "\f122";
993 }
994 .fa-star-half-empty:before,
995 .fa-star-half-full:before,
996 .fa-star-half-o:before {
997 content: "\f123";
998 }
999 .fa-location-arrow:before {
1000 content: "\f124";
1001 }
1002 .fa-crop:before {
1003 content: "\f125";
1004 }
1005 .fa-code-fork:before {
1006 content: "\f126";
1007 }
1008 .fa-unlink:before,
1009 .fa-chain-broken:before {
1010 content: "\f127";
1011 }
1012 .fa-question:before {
1013 content: "\f128";
1014 }
1015 .fa-info:before {
1016 content: "\f129";
1017 }
1018 .fa-exclamation:before {
1019 content: "\f12a";
1020 }
1021 .fa-superscript:before {
1022 content: "\f12b";
1023 }
1024 .fa-subscript:before {
1025 content: "\f12c";
1026 }
1027 .fa-eraser:before {
1028 content: "\f12d";
1029 }
```

```
1030 .fa-puzzle-piece:before {
1031   content: "\f12e";
1032 }
1033 .fa-microphone:before {
1034   content: "\f130";
1035 }
1036 .fa-microphone-slash:before {
1037   content: "\f131";
1038 }
1039 .fa-shield:before {
1040   content: "\f132";
1041 }
1042 .fa-calendar-o:before {
1043   content: "\f133";
1044 }
1045 .fa-fire-extinguisher:before {
1046   content: "\f134";
1047 }
1048 .fa-rocket:before {
1049   content: "\f135";
1050 }
1051 .fa-maxcdn:before {
1052   content: "\f136";
1053 }
1054 .fa-chevron-circle-left:before {
1055   content: "\f137";
1056 }
1057 .fa-chevron-circle-right:before {
1058   content: "\f138";
1059 }
1060 .fa-chevron-circle-up:before {
1061   content: "\f139";
1062 }
1063 .fa-chevron-circle-down:before {
1064   content: "\f13a";
1065 }
1066 .fa-html5:before {
1067   content: "\f13b";
1068 }
1069 .fa-css3:before {
1070   content: "\f13c";
1071 }
1072 .fa-anchor:before {
1073   content: "\f13d";
1074 }
1075 .fa-unlock-alt:before {
1076   content: "\f13e";
1077 }
1078 .fa-bullseye:before {
1079   content: "\f140";
1080 }
1081 .fa-ellipsis-h:before {
1082   content: "\f141";
1083 }
1084 .fa-ellipsis-v:before {
1085   content: "\f142";
1086 }
1087 .fa-rss-square:before {
1088   content: "\f143";
1089 }
1090 .fa-play-circle:before {
1091   content: "\f144";
1092 }
1093 .fa-ticket:before {
1094   content: "\f145";
1095 }
1096 .fa-minus-square:before {
1097   content: "\f146";
1098 }
1099 .fa-minus-square-o:before {
1100   content: "\f147";
1101 }
1102 .fa-level-up:before {
1103   content: "\f148";
1104 }
1105 .fa-level-down:before {
1106   content: "\f149";
1107 }
1108 .fa-check-square:before {
1109   content: "\f14a";
1110 }
1111 .fa-pencil-square:before {
1112   content: "\f14b";
1113 }
1114 .fa-external-link-square:before {
1115   content: "\f14c";
1116 }
1117 .fa-share-square:before {
1118   content: "\f14d";
1119 }
1120 .fa-compass:before {
1121   content: "\f14e";
1122 }
1123 .fa-toggle-down:before
1124 .fa-caret-square-o-down:before {
1125   content: "\f150";
1126 }
1127 .fa-toggle-up:before
1128 .fa-caret-square-o-up:before {
1129   content: "\f151";
1130 }
1131 .fa-toggle-right:before
1132 .fa-caret-square-o-right:before {
1133   content: "\f152";
1134 }
1135 .fa-euro:before
1136 .fa-eur:before {
1137   content: "\f153";
1138 }
1139 .fa-gbp:before {
1140   content: "\f154";
1141 }
1142 .fa-dollar:before
1143 .fa-usd:before {
1144   content: "\f155";
```

```
1145 }
1146 .fa-rupee:before
1147 .fa-inr:before {
1148   content: "\f156";
1149 }
1150 .fa-cny:before
1151 .fa-rmb:before
1152 .fa-yen:before
1153 .fa-jpy:before {
1154   content: "\f157";
1155 }
1156 .fa-ruble:before
1157 .fa-rouble:before
1158 .fa-rub:before {
1159   content: "\f158";
1160 }
1161 .fa-won:before
1162 .fa-krw:before {
1163   content: "\f159";
1164 }
1165 .fa-bitcoin:before
1166 .fa-btc:before {
1167   content: "\f15a";
1168 }
1169 .fa-file:before {
1170   content: "\f15b";
1171 }
1172 .fa-file-text:before {
1173   content: "\f15c";
1174 }
1175 .fa-sort-alpha-asc:before {
1176   content: "\f15d";
1177 }
1178 .fa-sort-alpha-desc:before {
1179   content: "\f15e";
1180 }
1181 .fa-sort-amount-asc:before {
1182   content: "\f160";
1183 }
1184 .fa-sort-amount-desc:before {
1185   content: "\f161";
1186 }
1187 .fa-sort-numeric-asc:before {
1188   content: "\f162";
1189 }
1190 .fa-sort-numeric-desc:before {
1191   content: "\f163";
1192 }
1193 .fa-thumbs-up:before {
1194   content: "\f164";
1195 }
1196 .fa-thumbs-down:before {
1197   content: "\f165";
1198 }
1199 .fa-youtube-square:before {
1200   content: "\f166";
1201 }
1202 .fa-youtube:before {
1203   content: "\f167";
1204 }
1205 .fa-xing:before {
1206   content: "\f168";
1207 }
1208 .fa-xing-square:before {
1209   content: "\f169";
1210 }
1211 .fa-youtube-play:before {
1212   content: "\f16a";
1213 }
1214 .fa-dropbox:before {
1215   content: "\f16b";
1216 }
1217 .fa-stack-overflow:before {
1218   content: "\f16c";
1219 }
1220 .fa-instagram:before {
1221   content: "\f16d";
1222 }
1223 .fa-flickr:before {
1224   content: "\f16e";
1225 }
1226 .fa-adn:before {
1227   content: "\f170";
1228 }
1229 .fa-bitbucket:before {
1230   content: "\f171";
1231 }
1232 .fa-bitbucket-square:before {
1233   content: "\f172";
1234 }
1235 .fa-tumblr:before {
1236   content: "\f173";
1237 }
1238 .fa-tumblr-square:before {
1239   content: "\f174";
1240 }
1241 .fa-long-arrow-down:before {
1242   content: "\f175";
1243 }
1244 .fa-long-arrow-up:before {
1245   content: "\f176";
1246 }
1247 .fa-long-arrow-left:before {
1248   content: "\f177";
1249 }
1250 .fa-long-arrow-right:before {
1251   content: "\f178";
1252 }
1253 .fa-apple:before {
1254   content: "\f179";
1255 }
1256 .fa-windows:before {
1257   content: "\f17a";
1258 }
1259 .fa-android:before {
```

```
1260 content: "\f17b";
1261 }
1262 .fa-linux:before {
1263 content: "\f17c";
1264 }
1265 .fa-dribbble:before {
1266 content: "\f17d";
1267 }
1268 .fa-skype:before {
1269 content: "\f17e";
1270 }
1271 .fa-foursquare:before {
1272 content: "\f180";
1273 }
1274 .fa-trello:before {
1275 content: "\f181";
1276 }
1277 .fa-female:before {
1278 content: "\f182";
1279 }
1280 .fa-male:before {
1281 content: "\f183";
1282 }
1283 .fa-gittip:before {
1284 content: "\f184";
1285 }
1286 .fa-sun-o:before {
1287 content: "\f185";
1288 }
1289 .fa-moon-o:before {
1290 content: "\f186";
1291 }
1292 .fa-archive:before {
1293 content: "\f187";
1294 }
1295 .fa-bug:before {
1296 content: "\f188";
1297 }
1298 .fa-vk:before {
1299 content: "\f189";
1300 }
1301 .fa-weibo:before {
1302 content: "\f18a";
1303 }
1304 .fa-renren:before {
1305 content: "\f18b";
1306 }
1307 .fa-pagelines:before {
1308 content: "\f18c";
1309 }
1310 .fa-stack-exchange:before {
1311 content: "\f18d";
1312 }
1313 .fa-arrow-circle-o-right:before {
1314 content: "\f18e";
1315 }
1316 .fa-arrow-circle-o-left:before {
1317 content: "\f190";
1318 }
1319 .fa-toggle-left:before
1320 .fa-caret-square-o-left:before {
1321 content: "\f191";
1322 }
1323 .fa-dot-circle-o:before {
1324 content: "\f192";
1325 }
1326 .fa-wheelchair:before {
1327 content: "\f193";
1328 }
1329 .fa-vimeo-square:before {
1330 content: "\f194";
1331 }
1332 .fa-turkish-lira:before
1333 .fa-try:before {
1334 content: "\f195";
1335 }
1336 .fa-plus-square-o:before {
1337 content: "\f196";
1338 }
```

**File:** ./static/rest\_framework/css/default.css



```

1  /* The navbar is fixed at >= 980px wide, so add padding to the body to prevent
2  content running up underneath it. */
3
4  h1 {
5    font-weight: 300;
6  }
7
8  h2, h3 {
9    font-weight: 300;
10 }
11
12 .resource-description, .response-info {
13   margin-bottom: 2em;
14 }
15
16 .version:before {
17   content: "v";
18   opacity: 0.6;
19   padding-right: 0.25em;
20 }
21
22 .version {
23   font-size: 70%;
24 }
25
26 .format-option {
27   font-family: Menlo, Consolas, "Andale Mono", "Lucida Console", monospace;
28 }
29
30 .button-form {
31   float: right;
32   margin-right: 1em;
33 }
34
35 td.nested {
36   padding: 0 !important;
37 }
38
39 td.nested > table {
40   margin: 0;
41 }
42
43 form select, form input:not([type=checkbox]), form textarea {
44   width: 90%;
45 }
46
47 form select[multiple] {
48   height: 150px;
49 }
50
51 /* To allow tooltips to work on disabled elements */
52 .disabled-tooltip-shield {
53   position: absolute;
54   top: 0;
55   right: 0;
56   bottom: 0;
57   left: 0;
58 }
59
60 .errorlist {
61   margin-top: 0.5em;
62 }
63
64 pre {
65   overflow: auto;
66   word-wrap: normal;
67   white-space: pre;
68   font-size: 12px;
69 }
70
71 .page-header {
72   border-bottom: none;
73   padding-bottom: 0px;
74 }
75
76 #filtersModal form input[type=submit] {
77   width: auto;
78 }
79
80 #filtersModal .modal-body h2 {
81   margin-top: 0;
82 }

```

**File:** `./static/rest_framework/css/prettify.css`

```

1  .com { color: #93a1a1; }
2  .lit { color: #195f91; }
3  .pun, .opn, .clo { color: #93a1a1; }
4  .fun { color: #dc322f; }
5  .str, .atv { color: #D14; }
6  .kwd, .pre, .print, .tag { color: #1e347b; }
7  .typ, .atn, .dec, .var { color: teal; }
8  .pln { color: #48484c; }
9
10 pre {
11   padding: 8px;
12   background-color: #f7f7f9;
13   border: 1px solid #e1e1e8;
14 }
15 pre {
16   padding-left: 10px;
17   padding-right: 10px;
18   padding-top: 10px;
19   padding-bottom: 10px;
20 }
21
22 /* Specify class=linenums on a pre to get line numbering */
23 pre .linenums {
24   margin: 0 0 33px 0; /* IE indents via margin-left */
25 }
26
27 pre .linenums li {
28   padding-left: 12px;
29   color: #bebec5;
30   line-height: 20px;
31   text-shadow: 0 1px 0 #fff;
32 }

```

**File: ./static/rest\_framework/css/bootstrap.min.css**

```
1 /*!
2 * Bootstrap v3.4.0 (https://getbootstrap.com/)
3 * Copyright 2011-2018 Twitter, Inc.
4 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
5 */
6 /* normalize.css v3.0.3 | MIT License | github.com/necolas/normalize.css */
7 html {font-family: sans-serif; -ms-text-size-adjust: 100%; -webkit-text-size-adjust: 100%;}
8 body {margin: 0;}
9 article, aside, details, main, menu, nav, section, summary {display: inline-block; vertical-align: top; width: 100%;}
```

**File:** `./static/rest_framework/css/bootstrap-tweaks.css`

```

2  /*
3  This CSS file contains some tweaks specific to the included Bootstrap theme.
4  It's separate from `style.css` so that it can be easily overridden by replacing
5  a single block in the template.
6
7  */
8
9  .form-actions {
10     background: transparent;
11     border-top-color: transparent;
12     padding-top: 0;
13     text-align: right;
14 }
15
16 #generic-content-form textarea {
17     font-family: Consolas, Monaco, Lucida Console, Liberation Mono, DejaVu Sans Mono, Bitstream Vera Sans Mono, Courier New, monospace;
18     font-size: 80%;
19 }
20
21 .navbar-inverse .brand a {
22     color: #999999;
23 }
24 .navbar-inverse .brand: hover a {
25     color: white;
26     text-decoration: none;
27 }
28
29 /* custom navigation styles */
30 .navbar {
31     width: 100%;
32     position: fixed;
33     left: 0;
34     top: 0;
35 }
36
37 .navbar {
38     background: #2C2C2C;
39     color: white;
40     border: none;
41     border-top: 5px solid #A30000;
42     border-radius: 0px;
43 }
44
45 .navbar .nav li, .navbar .nav li a, .navbar .brand: hover {
46     color: white;
47 }
48
49 .nav-list > .active > a, .nav-list > .active > a: hover {
50     background: #2C2C2C;
51 }
52
53 .navbar .dropdown-menu li a, .navbar .dropdown-menu li {
54     color: #A30000;
55 }
56
57 .navbar .dropdown-menu li a: hover {
58     background: #EEEEEE;
59     color: #C20000;
60 }
61
62 ul.breadcrumb {
63     margin: 70px 0 0 0;
64 }
65
66 .breadcrumb li.active a {
67     color: #777;
68 }

```

```

69
70 .pagination>.disabled>a,
71 .pagination>.disabled>a:hover,
72 .pagination>.disabled>a:focus {
73   cursor: not-allowed;
74   pointer-events: none;
75 }
76
77 .pager>.disabled>a,
78 .pager>.disabled>a:hover,
79 .pager>.disabled>a:focus {
80   pointer-events: none;
81 }
82
83 .pager.next {
84   margin-left: 10px;
85 }
86
87 /*=== dabapps bootstrap styles ===*/
88
89 html {
90   width: 100%;
91   background: none;
92 }
93
94 /*body, .navbar, .container-fluid {
95   max-width: 1150px;
96   margin: 0 auto;
97 }*/
98
99 body {
100   background: url("../img/grid.png") repeat-x;
101   background-attachment: fixed;
102 }
103
104 #content {
105   margin: 0;
106   padding-bottom: 60px;
107 }
108
109 /* sticky footer and footer */
110 html, body {
111   height: 100%;
112 }
113
114 .wrapper {
115   position: relative;
116   top: 0;
117   left: 0;
118   padding-top: 60px;
119   margin: -60px 0;
120   min-height: 100%;
121 }
122
123 form-switcher {
124   margin-bottom: 0;
125 }
126
127 well {
128   -webkit-box-shadow: none;
129   -moz-box-shadow: none;
130   box-shadow: none;
131 }
132
133 .well .form-actions {
134   padding-bottom: 0;
135   margin-bottom: 0;
136 }
137
138 .well form {
139   margin-bottom: 0;
140 }
141
142 nav-tabs {
143   border: 0;
144 }
145
146 .nav-tabs > li {
147   float: right;
148 }
149
150 .nav-tabs li a {
151   margin-right: 0;
152 }
153
154 .nav-tabs > .active > a {
155   background: #F5F5F5;
156 }
157
158 .nav-tabs > .active > a:hover {
159   background: #F5F5F5;
160 }
161
162 .tabbable.first-tab-active .tab-content {
163   border-top-right-radius: 0;
164 }
165
166 footer {
167   position: absolute;
168   bottom: 0;
169   left: 0;
170   clear: both;
171   z-index: 10;
172   height: 60px;
173   width: 95%;
174   margin: 0 2.5%;
175 }
176
177 footer p {
178   text-align: center;
179   color: gray;
180   border-top: 1px solid #DDDDDD;
181   padding-top: 10px;
182 }
183

```

```

184 footer a {
185   color: gray !important;
186   font-weight: bold;
187 }
188
189 footer a:hover {
190   color: gray;
191 }
192
193 .page-header {
194   border-bottom: none;
195   padding-bottom: 0px;
196   margin: 0;
197 }
198
199 /* custom general page styles */
200 .hero-unit h1, .hero-unit h2 {
201   color: #A30000;
202 }
203
204 body a {
205   color: #A30000;
206 }
207
208 body a:hover {
209   color: #c20000;
210 }
211
212 .request-info {
213   clear: both;
214 }
215
216 .horizontal-checkbox label {
217   padding-top: 0;
218 }
219
220 .horizontal-checkbox label {
221   padding-top: 0 !important;
222 }
223
224 .horizontal-checkbox input {
225   float: left;
226   width: 20px;
227   margin-top: 3px;
228 }
229
230 .modal-footer form {
231   margin-left: 5px;
232   margin-right: 5px;
233 }

```

**File: ./static/rest\_framework/css/bootstrap-theme.min.css**

```

1 /*!
2  * Bootstrap v3.4.0 (https://getbootstrap.com/)
3  * Copyright 2011-2018 Twitter, Inc.
4  * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
5  */
6 /*# sourceMappingURL=bootstrap-theme.min.css.map */

```

**File: ./static/rest\_framework/fonts/glyphicons-halflings-regular.svg**

[binary]

**File: ./static/rest\_framework/fonts/glyphicons-halflings-regular.woff**

[binary]

**File: ./static/rest\_framework/fonts/fontawesome-webfont.woff**

[binary]

**File: ./static/rest\_framework/fonts/glyphicons-halflings-regular.ttf**

[binary]

**File: ./static/rest\_framework/fonts/glyphicons-halflings-regular.eot**

[binary]

**File: ./static/rest\_framework/fonts/fontawesome-webfont.ttf**

[binary]

**File: ./static/rest\_framework/fonts/glyphicons-halflings-regular.woff2**

[binary]

**File: ./static/rest\_framework/fonts/fontawesome-webfont.svg**

[binary]

**File: ./static/rest\_framework/fonts/fontawesome-webfont.eot**

[binary]

**File: ./static/rest\_framework/js/default.js**

```
1 $(document).ready(function() {
2   //JSON highlighting.
3   prettyPrint();
4
5   // Bootstrap tooltips.
6   $('[js-tooltip]').tooltip({
7     delay: 1000,
8     container: 'body'
9   });
10
11  // Deal with rounded tab styling after tab clicks.
12  $('[data-toggle="tab"]:first').on('shown', function(e) {
13    $(e.target).parents('.tabbable').addClass('first-tab-active');
14  });
15
16  $('[data-toggle="tab"]:not(:first)').on('shown', function(e) {
17    $(e.target).parents('.tabbable').removeClass('first-tab-active');
18  });
19
20  $('[data-toggle="tab"]').click(function() {
21    document.cookie = "tabstyle=" + this.name + "; path=/";
22  });
23
24  // Store tab preference in cookies & display appropriate tab on load.
25  var selectedTab = null;
26  var selectedTabName = getCookie('tabstyle');
27
28  if (selectedTabName) {
29    selectedTabName = selectedTabName.replace(/[^a-z-]/g, "");
30  }
31
32  if (selectedTabName) {
33    selectedTab = $('[form-switcher a[name=' + selectedTabName + ']]');
34  }
35
36  if (selectedTab && selectedTab.length > 0) {
37    // Display whichever tab is selected.
38    selectedTab.tab('show');
39  } else {
40    // If no tab selected, display rightmost tab.
41    $('[form-switcher a:first]').tab('show');
42  }
43
44  $(window).on('load', function() {
45    $('#errorModal').modal('show');
46  });
47 });
```

**File: ./static/rest\_framework/js/jquery-3.5.1.min.js**

```
1 /*! jQuery v3.5.1 | (c) JS Foundation and other contributors | jquery.org/license */
2 (function(e,t){("use strict";"object"==typeof module&&"object"==typeof module.exports?module.exports=e:document?t(e,!0):function(e){if(!e)document.throw new Error("jQuery requires a window")
```

**File: ./static/rest\_framework/js/csrf.js**

```

1 function getCookie(name) {
2   var cookieValue = null;
3
4   if (document.cookie && document.cookie != "") {
5     var cookies = document.cookie.split(';');
6
7     for (var i = 0; i < cookies.length; i++) {
8       var cookie = jQuery.trim(cookies[i]);
9
10      // Does this cookie string begin with the name we want?
11      if (cookie.substring(0, name.length + 1) == (name + '=')) {
12        cookieValue = decodeURIComponent(cookie.substring(name.length + 1));
13        break;
14      }
15    }
16  }
17
18  return cookieValue;
19 }
20
21 function csrfSafeMethod(method) {
22   // these HTTP methods do not require CSRF protection
23   return /^(GET|HEAD|OPTIONS|TRACE)$/ test:method);
24 }
25
26 function sameOrigin(url) {
27   // test that a given url is a same-origin URL
28   // url could be relative or scheme relative or absolute
29   var host = document.location.host; // host + port
30   var protocol = document.location.protocol;
31   var sr_origin = '//' + host;
32   var origin = protocol + sr_origin;
33
34   // Allow absolute or scheme relative URLs to same origin
35   return (url == origin || url.slice(0, origin.length + 1) == origin + '/') ||
36     (url == sr_origin || url.slice(0, sr_origin.length + 1) == sr_origin + '/') ||
37     // or any other URL that isn't scheme relative or absolute i.e relative.
38     !/^(V|http|https:).*/ test:url);
39 }
40
41 var csrftoken = window.drf.csrfToken;
42
43 $.ajaxSetup({
44   beforeSend: function(xhr, settings) {
45     if (!csrfSafeMethod(settings.type) && sameOrigin(settings.url)) {
46       // Send the token to same-origin, relative URLs only.
47       // Send the token only if the method warrants CSRF protection
48       // Using the CSRFToken value acquired earlier
49       xhr.setRequestHeader(window.drf.csrfHeaderName, csrftoken);
50     }
51   }
52 });

```

#### File: ./static/rest\_framework/js/bootstrap.min.js

```

1 /*!
2  * Bootstrap v3.4.1 (https://getbootstrap.com/)
3  * Copyright 2011-2019 Twitter, Inc.
4  * Licensed under the MIT license
5  */
6 if("undefined"!==typeof jQuery)throw new Error("Bootstrap's JavaScript requires jQuery");function(t){t["use strict"];var e=jQuery.fn.jquery.split(" ")[0].split(".");if(e[0]<2&&e[1]<9||1==e[0]&&9==e[1]

```

#### File: ./static/rest\_framework/js/ajax-form.js

```

1 function replaceDocument(docString) {
2   var doc = document.open("text/html");
3
4   doc.write(docString);
5   doc.close();
6 }
7
8 function doAjaxSubmit(e) {
9   var form = $(this);
10  var btn = $(this.clt);
11  var method = (
12    btn.data('method') ||
13    form.data('method') ||
14    form.attr('method') || 'GET'
15  ).toUpperCase();
16
17  if (method === 'GET') {
18    // GET requests can always use standard form submits.
19    return;
20  }
21
22  var contentType =
23    form.find('input[data-override="content-type"]').val() ||
24    form.find('select[data-override="content-type"] option:selected').text();
25
26  if (method === 'POST' && !contentType) {
27    // POST requests can use standard form submits, unless we have
28    // overridden the content type.
29    return;
30  }
31
32  // At this point we need to make an AJAX form submission.
33  e.preventDefault();
34
35  var url = form.attr('action');
36  var data =
37
38  if (contentType) {
39    data = form.find('[data-override="content"]') .val() || ""
40
41    if (contentType === 'multipart/form-data') {
42      // We need to add a boundary parameter to the header
43      // We assume the first valid-looking boundary line in the body is correct
44      // regex is from RFC 2046 appendix A
45      var boundaryCharNoSpace = "0-9A-Z()+_-./:=?";

```

```

46     var boundaryChar = boundaryCharNoSpace + ' ';
47     var re = new RegExp('^--([' + boundaryChar + ']{0,69})[' + boundaryCharNoSpace + '])[\s]*?$', 'im');
48     var boundary = data.match(re);
49     if (boundary !== null) {
50         contentType += '; boundary="' + boundary[1] + '"';
51     }
52     // Fix textarea.value EOL normalisation (multipart/form-data should use CR+NL, not NL)
53     data = data.replace(/\n/g, '\r\n');
54 }
55 } else {
56     contentType = form.attr('enctype') || form.attr('encoding')
57 }
58 if (contentType === 'multipart/form-data') {
59     if (!window.FormData) {
60         alert('Your browser does not support AJAX multipart form submissions');
61         return;
62     }
63 }
64 // Use the FormData API and allow the content type to be set automatically,
65 // so it includes the boundary string.
66 // See https://developer.mozilla.org/en-US/docs/Web/API/FormData/Using_FormData_Objects
67 contentType = false;
68 data = new FormData(form[0]);
69 } else {
70     contentType = 'application/x-www-form-urlencoded; charset=UTF-8'
71     data = form.serialize();
72 }
73 }
74 }
75 var ret = $.ajax({
76     url: url,
77     method: method,
78     data: data,
79     contentType: contentType,
80     processData: false,
81     headers: {
82         'Accept': 'text/html; q=1.0, */*'
83     },
84 });
85
86 ret.always(function(data, textStatus, jqXHR) {
87     if (textStatus !== 'success') {
88         jqXHR = data;
89     }
90 }
91
92 var responseContentType = jqXHR.getResponseHeader("content-type") || "";
93
94 if (responseContentType.toLowerCase().indexOf('text/html') === 0) {
95     replaceDocument(jqXHR.responseText);
96 }
97
98 try {
99     // Modify the location and scroll to top, as if after page load.
100     history.replaceState({}, "", url);
101     scroll(0, 0);
102 } catch (err) {
103     // History API not supported, so redirect.
104     window.location = url;
105 }
106 } else {
107     // Not HTML content. We can't open this directly, so redirect.
108     window.location = url;
109 }
110 });
111
112 return ret;
113 }
114
115 function captureSubmittingElement(e) {
116     var target = e.target;
117     var form = this;
118     form.cik = target;
119 }
120
121 $.fn.ajaxForm = function() {
122     var options = {}
123     return this
124         .unbind('submit.form-plugin click.form-plugin')
125         .bind('submit.form-plugin', options.doAjaxSubmit)
126         .bind('click.form-plugin', options.captureSubmittingElement);
127 };

```

**File:** ./static/rest\_framework/js/prettify-min.js

[illegible]

```

1 (function(f){if(typeof exports==="object"&&typeof module!=="undefined"){module.exports=f()}else if(typeof define==="function"&&define.amd){define([],f)}else{var g;if(typeof window!=""
2 'use strict';
3
4 var _createClass = function () { function defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descriptor = props[i]; descriptor.enumerable = descriptor.enumerable || fal
5
6 function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }
7
8 var BasicAuthentication = function () {
9   function BasicAuthentication() {
10     var options = arguments.length > 0 && arguments[0] !== undefined ? arguments[0] : {};
11
12     _classCallCheck(this, BasicAuthentication);
13
14     var username = options.username;
15     var password = options.password;
16     var hash = window.btoa(username + ':' + password);
17     this.auth = 'Basic ' + hash;
18   }
19
20   _createClass(BasicAuthentication, [{
21     key: 'authenticate',
22     value: function authenticate(options) {
23       options.headers['Authorization'] = this.auth;
24       return options;
25     }
26   }]);
27
28   return BasicAuthentication;
29 }();
30
31 module.exports = {
32   BasicAuthentication: BasicAuthentication
33 };
34
35 }, {}).2:(function(require,module,exports){
36 'use strict';
37
38 var basic = require('./basic');
39 var session = require('./session');
40 var token = require('./token');
41
42 module.exports = {
43   BasicAuthentication: basic.BasicAuthentication,
44   SessionAuthentication: session.SessionAuthentication,
45   TokenAuthentication: token.TokenAuthentication
46 };
47
48 },{"/basic":1,"./session":3,"./token":4}].3:(function(require,module,exports){
49 'use strict';
50
51 var _createClass = function () { function defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descriptor = props[i]; descriptor.enumerable = descriptor.enumerable || fal
52
53 function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }
54
55 var utils = require('./utils');
56
57 function trim(str) {
58   return str.replace(/^\s\s*/, '').replace(/\s\s$/, '');
59 }
60
61 function getCookie(cookieName, cookieString) {
62   cookieString = cookieString || window.document.cookie;
63   if (cookieString && cookieString.indexOf(cookieName + "=") !== -1) {
64     var cookies = cookieString.split(';');
65     for (var i = 0; i < cookies.length; i++) {
66       var cookie = trim(cookies[i]);
67       // Does this cookie string begin with the name we want?
68       if (cookie.substr(0, cookieName.length + 1) === cookieName + '=') {
69         return decodeURIComponent(cookie.substr(cookieName.length + 1));
70       }
71     }
72   }
73   return null;
74 }
75
76 var SessionAuthentication = function () {
77   function SessionAuthentication() {
78     var options = arguments.length > 0 && arguments[0] !== undefined ? arguments[0] : {};
79
80     _classCallCheck(this, SessionAuthentication);
81
82

```

```
54  
55   var utils = require('./utils');  
56
```

```
57 function trim str {
58   return str replace(/^s\s*/,"") replace(/s\s*$/,"");
59 }
60 }
```

```
60
61 function getCookie(cookieName, cookieString) {
62   cookieString = cookieString || window.document.cookie;
63   if (cookieString.indexOf(cookieName) === -1) {
```

```
63   if (cookieString.indexOf(cookieString) !== -1) {
64     var cookies = cookieString.split(';');
65     for (var i = 0; i < cookies.length; i++) {
66       var cookie = trim(cookies[i]);
```

```
67 // Does this cookie string begin with the name we want?
68 if (cookie.substring(0, cookieName.length + 1) === cookieName + '=') {
69     return decodeURIComponent(cookie.substring(cookieName.length + 1));

```

```
76 var SessionAuthentication = function () {
77     function SessionAuthentication() {
78         var options = arguments.length > 0 && arguments[0] !== undefined ? arguments[0] : {};
79     }
80 }
```

```
79  
80     _classCallCheck(this, SessionAuthentication);  
81
```



```

82   this.csrfToken = getCookie(options.csrfCookieName, options.cookieString);
83   this.csrfHeaderName = options.csrfHeaderName;
84 }
85
86 _createClass(SessionAuthentication, [{
87   key: 'authenticate',
88   value: function authenticate(options) {
89     options.credentials = 'same-origin';
90     if (this.csrfToken && !utils.csrfSafeMethod(options.method)) {
91       options.headers[this.csrfHeaderName] = this.csrfToken;
92     }
93     return options;
94   }
95 }]);
96
97 return SessionAuthentication;
98 }();
99
100 module.exports = {
101   SessionAuthentication: SessionAuthentication
102 };
103
104 }, {"./utils":15}],4,{function(require,module,exports){
105   'use strict';
106
107   var _createClass = function () { function defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descriptor = props[i]; descriptor.enumerable = descriptor.enumerable || false;
108   function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }
109   var TokenAuthentication = function () {
110     function TokenAuthentication() {
111       var options = arguments.length > 0 && arguments[0] !== undefined ? arguments[0] : {};
112       _classCallCheck(this, TokenAuthentication);
113
114       this.token = options.token;
115       this.scheme = options.scheme || 'Bearer';
116     }
117
118     _createClass(TokenAuthentication, [{
119       key: 'authenticate',
120       value: function authenticate(options) {
121         options.headers['Authorization'] = this.scheme + ' ' + this.token;
122         return options;
123       }
124 }]);
125
126 return TokenAuthentication;
127 }();
128
129 module.exports = {
130   TokenAuthentication: TokenAuthentication
131 };
132
133 }, {"./utils":15}],5,{function(require,module,exports){
134   'use strict';
135
136   var _createClass = function () { function defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descriptor = props[i]; descriptor.enumerable = descriptor.enumerable || false;
137   function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }
138   var document = require('./document');
139   var codecs = require('./codecs');
140   var errors = require('./errors');
141   var transports = require('./transports');
142   var utils = require('./utils');
143
144   function lookupLink(node, keys) {
145     var _iteratorNormalCompletion = true;
146     var _didIteratorError = false;
147     var _iteratorError = undefined;
148
149     try {
150       for (var _iterator = keys[Symbol.iterator](), _step; !(_iteratorNormalCompletion = (_step = _iterator.next()) done); _iteratorNormalCompletion = true) {
151         var key = _step.value;
152
153         if (node instanceof document.Document) {
154           node = node.content[key];
155         } else {
156           node = node[key];
157         }
158
159         if (node === undefined) {
160           throw new errors.LinkLookupError('Invalid link lookup: ' + JSON.stringify(keys));
161         }
162       }
163     } catch (err) {
164       _didIteratorError = true;
165       _iteratorError = err;
166     } finally {
167       try {
168         if (!_iteratorNormalCompletion && _iterator.return) {
169           _iterator.return();
170         }
171       } finally {
172         if (_didIteratorError) {
173           throw _iteratorError;
174         }
175       }
176     }
177
178     if (!node instanceof document.Link) {
179       throw new errors.LinkLookupError('Invalid link lookup: ' + JSON.stringify(keys));
180     }
181
182     return node;
183   }
184
185   var Client = function () {
186     function Client() {
187       var options = arguments.length > 0 && arguments[0] !== undefined ? arguments[0] : {};
188       _classCallCheck(this, Client);
189
190       var transportOptions = {
191         auth: options.auth || null,
192         headers: options.headers || {}
193       };

```

```

197     requestCallback: options.requestCallback,
198     responseCallback: options.responseCallback
199   };
200
201   this decoders = options.decoders || [new codecs.CoreJSONCodec(), new codecs.JSONCodec(), new codecs.TextCodec()];
202   this transports = options.transports || [new transports.HTTPTransport(transportOptions)];
203 }
204
205 _createClass(Client, [{
206   key: 'action',
207   value: function action(document, keys) {
208     var params = arguments.length > 2 && arguments[2] !== undefined ? arguments[2] : {};
209
210     var link = lookupLink(document, keys);
211     var transport = utils.determineTransport(this.transports, link.url);
212     return transport.action(link, this.decoders, params);
213   },
214   {
215     key: 'get',
216     value: function get(url) {
217       var link = new document.Link(url, 'get');
218       var transport = utils.determineTransport(this.transports, url);
219       return transport.action(link, this.decoders);
220     }
221   }
222 ]]);
223
224 return Client;
225 }();
226
227 module.exports = {
228   Client: Client
229 };
230
231 }, {"/codecs":7,"./document":10,"./errors":11,"./transports":14,"./utils":15}];6:(function(require,module,exports){
232 'use strict';
233
234 var _createClass = function () { function defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descriptor = props[i]; descriptor.enumerable = descriptor.enumerable || false;
235   var _typeOf = typeof Symbol === "function" && typeof Symbol.iterator === "symbol" ? function (obj) { return typeof obj; } : function (obj) { return obj && typeof Symbol === "function" && c
236   function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }
237
238   var document = require('./document');
239   var URL = require('url-parse');
240
241   function unescapeKey(key) {
242     if (key.match(/_([type|meta])$/)) {
243       return key.substring(1);
244     }
245     return key;
246   }
247
248   function getString(obj, key) {
249     var value = obj[key];
250     if (typeof value === 'string') {
251       return value;
252     }
253     return "";
254   }
255
256   function getBoolean(obj, key) {
257     var value = obj[key];
258     if (typeof value === 'boolean') {
259       return value;
260     }
261     return false;
262   }
263
264   function getObject(obj, key) {
265     var value = obj[key];
266     if ((typeof value === 'undefined' ? 'undefined' : _typeof(value)) === 'object') {
267       return value;
268     }
269     return {};
270   }
271
272   function getArray(obj, key) {
273     var value = obj[key];
274     if (value instanceof Array) {
275       return value;
276     }
277     return [];
278   }
279
280   function getContent(data, baseUrl) {
281     var excluded = ['_type', '_meta'];
282     var content = {};
283     for (var property in data) {
284       if (data.hasOwnProperty(property) && !excluded.includes(property)) {
285         var key = unescapeKey(property);
286         var value = primitiveToNode(data[property], baseUrl);
287         content[key] = value;
288       }
289     }
290     return content;
291   }
292
293   function primitiveToNode(data, baseUrl) {
294     var isObject = data instanceof Object && !data instanceof Array;
295
296     if (isObject && data._type === 'document') {
297       // Document
298       var meta = getObject(data, '_meta');
299       var relativeUrl = getString(meta, 'url');
300       var url = relativeUrl ? URL(relativeUrl, baseUrl).toString() : "";
301       var title = getString(meta, 'title');
302       var description = getString(meta, 'description');
303       var content = getContent(data, url);
304       return new document.Document(url, title, description, content);
305     } else if (isObject && data._type === 'link') {
306       // Link
307       var _relativeUrl = getString(data, 'url');
308       var _url = _relativeUrl ? URL(_relativeUrl, baseUrl).toString() : "";
309       var method = getString(data, 'action') || 'get';
310       var _title = getString(data, 'title');

```

```

312     var _description = getString(data, 'description');
313     var fieldsData = getArray(data, 'fields');
314     var fields = [];
315     for (var idx = 0, len = fieldsData.length; idx < len; idx++) {
316         var value = fieldsData[idx];
317         var name = getString(value, 'name');
318         var required = getBoolean(value, 'required');
319         var location = getString(value, 'location');
320         var fieldDescription = getString(value, 'fieldDescription');
321         var field = new document.Field(name, required, location, fieldDescription);
322         fields.push(field);
323     }
324     return new document.Link(_url, method, 'application/json', fields, _title, _description);
325 } else if (isObject) {
326     // Object
327     var _content = {};
328     for (var key in data) {
329         if (data.hasOwnProperty(key)) {
330             _content[key] = primitiveToNode(data[key], baseUrl);
331         }
332     }
333     return _content;
334 } else if (data instanceof Array) {
335     // Object
336     var _content2 = [];
337     for (var _idx = 0, _len = data.length, _idx < _len, _idx++) {
338         _content2.push(primitiveToNode(data[_idx], baseUrl));
339     }
340     return _content2;
341 }
342 // Primitive
343 return data;
344 }
345
346 var CoreJSONCodec = function () {
347     function CoreJSONCodec() {
348         _classCallCheck(this, CoreJSONCodec);
349
350         this.mediaType = 'application/coreapi+json';
351     }
352
353     _createClass(CoreJSONCodec, [{
354         key: 'decode',
355         value: function decode(text) {
356             var options = arguments.length > 1 && arguments[1] !== undefined ? arguments[1] : {};
357
358             var data = text;
359             if (options.preloaded === undefined || options.preloaded) {
360                 data = JSON.parse(text);
361             }
362             return primitiveToNode(data, options.url);
363         }
364     }]);
365
366     return CoreJSONCodec;
367 }();
368
369 module.exports = {
370     CoreJSONCodec: CoreJSONCodec
371 };
372
373 }, {"./document":10,"url-parse":19}],7:{function(require,module,exports){
374     'use strict';
375
376     var corejson = require('./corejson');
377     var json = require('./json');
378     var text = require('./text');
379
380     module.exports = {
381         CoreJSONCodec: corejson.CoreJSONCodec,
382         JSONCodec: json.JSONCodec,
383         TextCodec: text.TextCodec
384     };
385
386 }, {"./corejson":6,"./json":8,"./text":9}],8:{function(require,module,exports){
387     'use strict';
388
389     var _createClass = function () { function defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descriptor = props[i]; descriptor.enumerable = descriptor.enumerable || false;
390
391     function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }
392
393     var JSONCodec = function () {
394         function JSONCodec() {
395             _classCallCheck(this, JSONCodec);
396
397             this.mediaType = 'application/json';
398         }
399
400         _createClass(JSONCodec, [{
401             key: 'decode',
402             value: function decode(text) {
403                 var options = arguments.length > 1 && arguments[1] !== undefined ? arguments[1] : {};
404
405                 return JSON.parse(text);
406             }
407         }]);
408
409         return JSONCodec;
410     }();
411
412     module.exports = {
413         JSONCodec: JSONCodec
414     };
415
416 }, {"./text":9}],9:{function(require,module,exports){
417     'use strict';
418
419     var _createClass = function () { function defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descriptor = props[i]; descriptor.enumerable = descriptor.enumerable || false;
420
421     function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }
422
423     var TextCodec = function () {
424         function TextCodec() {
425             _classCallCheck(this, TextCodec);
426

```

```

427   this.mediaType = 'text/';
428 }
429
430 _createClass(TextCodec, [{
431   key: 'decode',
432   value: function decode(text) {
433     var options = arguments.length > 1 && arguments[1] !== undefined ? arguments[1] : {};
434
435     return text;
436   }
437 }]);
438
439 return TextCodec;
440 }();
441
442 module.exports = {
443   TextCodec: TextCodec
444 };
445
446 },{}],10:[function(require,module,exports){
447   'use strict';
448
449   function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }
450
451   var Document = function Document() {
452     var url = arguments.length > 0 && arguments[0] !== undefined ? arguments[0] : "";
453     var title = arguments.length > 1 && arguments[1] !== undefined ? arguments[1] : "";
454     var description = arguments.length > 2 && arguments[2] !== undefined ? arguments[2] : "";
455     var content = arguments.length > 3 && arguments[3] !== undefined ? arguments[3] : {};
456
457     _classCallCheck(this, Document);
458
459     this.url = url;
460     this.title = title;
461     this.description = description;
462     this.content = content;
463   };
464
465   var Link = function Link(url, method) {
466     var encoding = arguments.length > 2 && arguments[2] !== undefined ? arguments[2] : 'application/json';
467     var fields = arguments.length > 3 && arguments[3] !== undefined ? arguments[3] : [];
468     var title = arguments.length > 4 && arguments[4] !== undefined ? arguments[4] : "";
469     var description = arguments.length > 5 && arguments[5] !== undefined ? arguments[5] : "";
470
471     _classCallCheck(this, Link);
472
473     if (url === undefined) {
474       throw new Error('url argument is required');
475     }
476
477     if (method === undefined) {
478       throw new Error('method argument is required');
479     }
480
481     this.url = url;
482     this.method = method;
483     this.encoding = encoding;
484     this.fields = fields;
485     this.title = title;
486     this.description = description;
487   };
488
489   var Field = function Field(name) {
490     var required = arguments.length > 1 && arguments[1] !== undefined ? arguments[1] : false;
491     var location = arguments.length > 2 && arguments[2] !== undefined ? arguments[2] : "";
492     var description = arguments.length > 3 && arguments[3] !== undefined ? arguments[3] : "";
493
494     _classCallCheck(this, Field);
495
496     if (name === undefined) {
497       throw new Error('name argument is required');
498     }
499
500     this.name = name;
501     this.required = required;
502     this.location = location;
503     this.description = description;
504   };
505
506   module.exports = {
507     Document: Document,
508     Link: Link,
509     Field: Field
510   };
511
512 },{}],11:[function(require,module,exports){
513   'use strict';
514
515   function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }
516
517   function _possibleConstructorReturn(self, call) { if (!self) { throw new ReferenceError("this hasn't been initialised - super() hasn't been called"); } return call && (typeof call === "object" || typeof
518
519   function _inherits(subClass, superClass) { if (typeof superClass !== "function" && superClass !== null) { throw new TypeError("Super expression must either be null or a function, not " + typeof
520
521   var ParameterError = function (_Error) {
522     _inherits(ParameterError, _Error);
523
524     function ParameterError(message) {
525       _classCallCheck(this, ParameterError);
526
527       var _this = _possibleConstructorReturn(this, (ParameterError.__proto__ || Object.getPrototypeOf(ParameterError)).call(this, message));
528
529       _this.message = message;
530       _this.name = 'ParameterError';
531       return _this;
532     }
533
534     return ParameterError;
535   }(Error);
536
537   var LinkLookupError = function (_Error2) {
538     _inherits(LinkLookupError, _Error2);
539
540     function LinkLookupError(message) {
541       _classCallCheck(this, LinkLookupError);

```

```

542
543     var _this2 = _possibleConstructorReturn(this, (LinkLookupError.__proto__ || Object.getPrototypeOf(LinkLookupError)).call(this, message));
544
545     _this2.message = message;
546     _this2.name = 'LinkLookupError';
547     return _this2;
548 }
549
550 return LinkLookupError;
551 }(Error);
552
553 var ErrorMessage = function (_Error3) {
554   _inherits(ErrorMessage, _Error3);
555
556   function ErrorMessage(message, content) {
557     _classCallCheck(this, ErrorMessage);
558
559     var _this3 = _possibleConstructorReturn(this, (ErrorMessage.__proto__ || Object.getPrototypeOf(ErrorMessage)).call(this, message));
560
561     _this3.message = message;
562     _this3.content = content;
563     _this3.name = 'ErrorMessage';
564     return _this3;
565   }
566
567   return ErrorMessage;
568 }(Error);
569
570 module.exports = {
571   ParameterError: ParameterError,
572   LinkLookupError: LinkLookupError,
573   ErrorMessage: ErrorMessage
574 };
575
576 }, {}], 12: (function (require, module, exports) {
577   'use strict';
578
579   var auth = require('./auth');
580   var client = require('./client');
581   var codecs = require('./codecs');
582   var document = require('./document');
583   var errors = require('./errors');
584   var transports = require('./transports');
585   var utils = require('./utils');
586
587   var coreapi = {
588     Client: client.Client,
589     Document: document.Document,
590     Link: document.Link,
591     auth: auth,
592     codecs: codecs,
593     errors: errors,
594     transports: transports,
595     utils: utils
596   };
597
598   module.exports = coreapi;
599
600   }, {"/auth": 2, "/client": 5, "/codecs": 7, "/document": 10, "/errors": 11, "/transports": 14, "/utils": 15}], 13: (function (require, module, exports) {
601     'use strict';
602
603     var _createClass = function () { function defineProperties(target, props) { for (var i = 0; i < props.length; i++) { var descriptor = props[i]; descriptor.enumerable = descriptor.enumerable || false;
604
605     function _classCallCheck(instance, Constructor) { if (!(instance instanceof Constructor)) { throw new TypeError("Cannot call a class as a function"); } }
606
607     var fetch = require('isomorphic-fetch');
608     var errors = require('./errors');
609     var utils = require('./utils');
610     var URL = require('url-parse');
611     var urlTemplate = require('url-template');
612
613     var parseResponse = function parseResponse(response, decoders, responseCallback) {
614       return response.text().then(function (text) {
615         if (responseCallback) {
616           responseCallback(response, text);
617         }
618         var contentType = response.headers.get('Content-Type');
619         var decoder = utils.negotiateDecoder(decoders, contentType);
620         var options = { url: response.url };
621         return decoder.decode(text, options);
622       });
623     };
624
625     var HTTPTransport = function () {
626       function HTTPTransport() {
627         var options = arguments.length > 0 && arguments[0] !== undefined ? arguments[0] : {};
628
629         _classCallCheck(this, HTTPTransport);
630
631         this.schemes = ['http', 'https'];
632         this.auth = options.auth || null;
633         this.headers = options.headers || {};
634         this.fetch = options.fetch || fetch;
635         this.FormData = options.FormData || window.FormData;
636         this.requestCallback = options.requestCallback;
637         this.responseCallback = options.responseCallback;
638       }
639
640       _createClass(HTTPTransport, [{
641         key: 'buildRequest',
642         value: function buildRequest(link, decoders) {
643           var params = arguments.length > 2 && arguments[2] !== undefined ? arguments[2] : {};
644
645           var fields = link.fields;
646           var method = link.method.toUpperCase();
647           var queryParams = {};
648           var pathParams = {};
649           var formParams = {};
650           var fieldNames = [];
651           var hasBody = false;
652
653           for (var idx = 0; len = fields.length, idx < len; idx++) {
654             var field = fields[idx];
655
656             // Ensure any required fields are included

```

```

657     if (!params.hasOwnProperty(field.name)) {
658         if (field.required) {
659             throw new errors.ParameterError('Missing required field: ' + field.name + '');
660         } else {
661             continue;
662         }
663     }
664
665     fieldNames.push(field.name);
666     if (field.location === 'query') {
667         queryParams[field.name] = params[field.name];
668     } else if (field.location === 'path') {
669         pathParams[field.name] = params[field.name];
670     } else if (field.location === 'form') {
671         formParams[field.name] = params[field.name];
672         hasBody = true;
673     } else if (field.location === 'body') {
674         formParams = params[field.name];
675         hasBody = true;
676     }
677 }
678
679 // Check for any parameters that did not have a matching field
680 for (var property in params) {
681     if (params.hasOwnProperty(property) && !fieldNames.includes(property)) {
682         throw new errors.ParameterError('Unknown parameter: ' + property + '');
683     }
684 }
685
686 var requestOptions = { method: method, headers: {} };
687
688 Object.assign(requestOptions.headers, this.headers);
689
690 if (hasBody) {
691     if (link.encoding === 'application/json') {
692         requestOptions.body = JSON.stringify(formParams);
693         requestOptions.headers['Content-Type'] = 'application/json';
694     } else if (link.encoding === 'multipart/form-data') {
695         var form = new this.FormData();
696
697         for (var paramKey in formParams) {
698             form.append(paramKey, formParams[paramKey]);
699         }
700         requestOptions.body = form;
701     } else if (link.encoding === 'application/x-www-form-urlencoded') {
702         var formBody = [];
703         for (var _paramKey in formParams) {
704             var encodedKey = encodeURIComponent(_paramKey);
705             var encodedValue = encodeURIComponent(formParams[_paramKey]);
706             formBody.push(encodedKey + '=' + encodedValue);
707         }
708         formBody = formBody.join('&');
709
710         requestOptions.body = formBody;
711         requestOptions.headers['Content-Type'] = 'application/x-www-form-urlencoded';
712     }
713 }
714
715 if (this.auth) {
716     requestOptions = this.auth.authenticate(requestOptions);
717 }
718
719 var parsedUrl = urlTemplate.parse(link.url);
720 parsedUrl = parsedUrl.expand(pathParams);
721 parsedUrl = new URL(parsedUrl);
722 parsedUrl.set('query', queryParams);
723
724 return {
725     url: parsedUrl.toString(),
726     options: requestOptions
727 };
728 }, {
729     key: 'action',
730     value: function action(link, decoders) {
731         var params = arguments.length > 2 && arguments[2] !== undefined ? arguments[2] : {};
732
733         var responseCallback = this.responseCallback;
734         var request = this.buildRequest(link, decoders, params);
735
736         if (this.requestCallback) {
737             this.requestCallback(request);
738         }
739
740         return this.fetch(request.url, request.options).then(function (response) {
741             if (response.status === 204) {
742                 return;
743             }
744             return parseResponse(response, decoders, responseCallback).then(function (data) {
745                 if (response.ok) {
746                     return data;
747                 } else {
748                     var title = response.status + ' ' + response.statusText;
749                     var error = new errors.ErrorMessage(title, data);
750                     return Promise.reject(error);
751                 }
752             });
753         });
754     }
755 });
756
757 return HTTPTransport;
758 }();
759
760 module.exports = {
761     HTTPTransport: HTTPTransport
762 };
763
764 }, {
765     './errors': 11, './utils': 15, 'isomorphic-fetch': 16, 'url-parse': 19, 'url-template': 21
766 }, 14, {function(require, module, exports) {
767     'use strict';
768     var http = require('http');
769
770     module.exports = {
771         HTTPTransport: http.HTTPTransport
772     };
773 }

```

```

772 };
773
774 },{"/http":13}],15:{function(require,module,exports){
775 'use strict';
776
777 var URL = require('url-parse');
778
779 var determineTransport = function determineTransport(transports, url) {
780   var parsedUrl = new URL(url);
781   var scheme = parsedUrl.protocol.replace(':', '');
782
783   var _iteratorNormalCompletion = true;
784   var _didIteratorError = false;
785   var _iteratorError = undefined;
786
787   try {
788     for (var _iterator = transports.Symbol.iterator[()], _step2; !(_iteratorNormalCompletion = (_step = _iterator.next()) done); _iteratorNormalCompletion = true) {
789       var transport = _step.value;
790
791       if (transport.schemes.includes(scheme)) {
792         return transport;
793       }
794     }
795   } catch (err) {
796     _didIteratorError = true;
797     _iteratorError = err;
798   } finally {
799     try {
800       if (!_iteratorNormalCompletion && _iterator.return) {
801         _iterator.return();
802       }
803     } finally {
804       if (_didIteratorError) {
805         throw _iteratorError;
806       }
807     }
808   }
809
810   throw Error('Unsupported scheme in URL: ' + url);
811 };
812
813 var negotiateDecoder = function negotiateDecoder(decoders, contentType) {
814   if (contentType === undefined || contentType === null) {
815     return decoders[0];
816   }
817
818   var fullType = contentType.toLowerCase().split(';')[0].trim();
819   var mainType = fullType.split('/')[0] + '/' + fullType.split('/')[1];
820   var wildcardType = '*/*';
821   var acceptableTypes = [fullType, mainType, wildcardType];
822
823   var _iteratorNormalCompletion2 = true;
824   var _didIteratorError2 = false;
825   var _iteratorError2 = undefined;
826
827   try {
828     for (var _iterator2 = decoders.Symbol.iterator[()], _step2; !(_iteratorNormalCompletion2 = (_step2 = _iterator2.next()) done); _iteratorNormalCompletion2 = true) {
829       var decoder = _step2.value;
830
831       if (acceptableTypes.includes(decoder.mediaType)) {
832         return decoder;
833       }
834     }
835   } catch (err) {
836     _didIteratorError2 = true;
837     _iteratorError2 = err;
838   } finally {
839     try {
840       if (!_iteratorNormalCompletion2 && _iterator2.return) {
841         _iterator2.return();
842       }
843     } finally {
844       if (_didIteratorError2) {
845         throw _iteratorError2;
846       }
847     }
848   }
849
850   throw Error('Unsupported media in Content-Type header: ' + contentType);
851 };
852
853 var csrfSafeMethod = function csrfSafeMethod(method) {
854   // these HTTP methods do not require CSRF protection
855   return /^(GET|HEAD|OPTIONS|TRACE)$/i.test(method);
856 };
857
858 module.exports = {
859   determineTransport: determineTransport,
860   negotiateDecoder: negotiateDecoder,
861   csrfSafeMethod: csrfSafeMethod
862 };
863
864 },{"/url-parse":19}],16:{function(require,module,exports){
865 // the whatwg-fetch polyfill installs the fetch() function
866 // on the global object (window or self)
867 //
868 // Return that as the export for use in Webpack, Browserify etc.
869 require('whatwg-fetch');
870 module.exports = self.fetch.bind(self);
871
872 },{"/whatwg-fetch":22}],17:{function(require,module,exports){
873 'use strict';
874
875 var has = Object.prototype.hasOwnProperty;
876
877 /**
878 * Simple query string parser.
879 *
880 * @param {String} query The query string that needs to be parsed.
881 * @returns {Object}
882 * @api public
883 */
884 function querystring(query) {
885   var parser = /(?:^&?&)+=?((?&)*)/g;

```

```

887     , result = {}
888     , part:
889
890     //
891     // Little nifty parsing hack, leverage the fact that RegExp.exec increments
892     // the lastIndex property so we can continue executing this loop until we've
893     // parsed all results.
894     //
895     for (;
896         part = parser.exec(query);
897         result[decodeURIComponent(part[1])] = decodeURIComponent(part[2])
898     );
899
900     return result;
901 }
902
903 /**
904  * Transform a query string to an object.
905  *
906  * @param {Object} obj Object that should be transformed.
907  * @param {String} prefix Optional prefix.
908  * @returns {String}
909  * @api public
910  */
911 function querystringify(obj, prefix) {
912     prefix = prefix || '';
913
914     var pairs = [];
915
916     //
917     // Optionally prefix with a '?' if needed
918     //
919     if ('string' !== typeof prefix) prefix = '?';
920
921     for (var key in obj) {
922         if (has.call(obj, key)) {
923             pairs.push(encodeURIComponent(key) + '=' + encodeURIComponent(obj[key]));
924         }
925     }
926
927     return pairs.length ? prefix + pairs.join('&') : '';
928 }
929
930 //
931 // Expose the module.
932 //
933 exports.stringify = querystringify;
934 exports.parse = querystring;
935
936 }, {}), 18: function(require, module, exports) {
937     'use strict';
938
939     /**
940     * Check if we're required to add a port number.
941     *
942     * @see https://url.spec.whatwg.org/#default-port
943     * @param {Number|String} port Port number we need to check
944     * @param {String} protocol Protocol we need to check against.
945     * @returns {Boolean} Is it a default port for the given protocol
946     * @api private
947     */
948     module.exports = function required(port, protocol) {
949         protocol = protocol.split(':')[0];
950         port = +port;
951
952         if (!port) return false;
953
954         switch (protocol) {
955             case 'http':
956             case 'ws':
957                 return port !== 80;
958
959             case 'https':
960             case 'wss':
961                 return port !== 443;
962
963             case 'ftp':
964                 return port !== 21;
965
966             case 'gopher':
967                 return port !== 70;
968
969             case 'file':
970                 return false;
971         }
972
973         return port !== 0;
974     };
975
976 }, {}), 19: function(require, module, exports) {
977     'use strict';
978
979     var required = require('requires-port')
980     , lolcation = require('lolcation')
981     , qs = require('querystringify')
982     , protocolre = /^(?![a-z][a-z0-9+~]*)(VV)?((\Ss)*)/i;
983
984     /**
985     * These are the parse rules for the URL parser, it informs the parser
986     * about:
987     *
988     * 0. The char it Needs to parse, if it's a string it should be done using
989     *    indexOf, RegExp using exec and NaN means set as current value.
990     * 1. The property we should set when parsing this value.
991     * 2. Indication if it's backwards or forward parsing, when set as number it's
992     *    the value of extra chars that should be split off.
993     * 3. Inherit from location if non existing in the parser.
994     * 4. `toLowerCase` the resulting value.
995     */
996     var rules = [
997         ['#', 'hash'], // Extract from the back.
998         ['?', 'query'], // Extract from the back.
999         ['/', 'pathname'], // Extract from the back.
1000        ['@', 'auth', 1], // Extract from the front.
1001        [NaN, 'host', undefined, 1, 1], // Set left over value.

```



```

1002 [/(d+)$/,'port',undefined,1], // RegExp the back.
1003 [NaN,'hostname',undefined,1,1] // Set left over.
1004 ];
1005
1006 /**
1007 * @typedef ProtocolExtract
1008 * @type Object
1009 * @property {String} protocol Protocol matched in the URL, in lowercase.
1010 * @property {Boolean} slashes `true` if protocol is followed by "://" , else `false`.
1011 * @property {String} rest Rest of the URL that is not part of the protocol.
1012 */
1013
1014 /**
1015 * Extract protocol information from a URL with/without double slash ("//").
1016 *
1017 * @param {String} address URL we want to extract from.
1018 * @return {ProtocolExtract} Extracted information.
1019 * @api private
1020 */
1021 function extractProtocol(address) {
1022   var match = protocolre.exec(address);
1023
1024   return {
1025     protocol: match[1] ? match[1].toLowerCase() : "",
1026     slashes: !!match[2],
1027     rest: match[3]
1028   };
1029 }
1030
1031 /**
1032 * Resolve a relative URL pathname against a base URL pathname.
1033 *
1034 * @param {String} relative Pathname of the relative URL.
1035 * @param {String} base Pathname of the base URL.
1036 * @return {String} Resolved pathname.
1037 * @api private
1038 */
1039 function resolve(relative, base) {
1040   var path = (base || '/').split('/').slice(0, -1).concat(relative.split('/'))
1041   , i = path.length
1042   , last = path[i - 1]
1043   , unshift = false
1044   , up = 0;
1045
1046   while (i--) {
1047     if (path[i] === '.') {
1048       path.splice(i, 1);
1049     } else if (path[i] === '..') {
1050       path.splice(i, 1);
1051       up++;
1052     } else if (up) {
1053       if (i === 0) unshift = true;
1054       path.splice(i, 1);
1055       up--;
1056     }
1057   }
1058
1059   if (unshift) path.unshift("");
1060   if (last === '.' || last === '..') path.push("");
1061   return path.join('/');
1062 }
1063
1064
1065 /**
1066 * The actual URL instance. Instead of returning an object we've opted-in to
1067 * create an actual constructor as it's much more memory efficient and
1068 * faster and it pleases my OCD.
1069 *
1070 * @constructor
1071 * @param {String} address URL we want to parse.
1072 * @param {Object|String} location Location defaults for relative paths.
1073 * @param {Boolean|Function} parser Parser for the query string.
1074 * @api public
1075 */
1076 function URL(address, location, parser) {
1077   if (!(this instanceof URL)) {
1078     return new URL(address, location, parser);
1079   }
1080
1081   var relative, extracted, parse, instruction, index, key;
1082   , instructions = rules.slice()
1083   , type = typeof location
1084   , url = this
1085   , i = 0;
1086
1087   //
1088   // The following if statements allows this module two have compatibility with
1089   // 2 different API:
1090   //
1091   // 1. Node.js's `url.parse` api which accepts a URL, boolean as arguments
1092   //    where the boolean indicates that the query string should also be parsed.
1093   //
1094   // 2. The `URL` interface of the browser which accepts a URL, object as
1095   //    arguments. The supplied object will be used as default values / fall-back
1096   //    for relative paths.
1097   //
1098   if ('object' !== type && 'string' !== type) {
1099     parser = location;
1100     location = null;
1101   }
1102
1103   if (parser && 'function' !== typeof parser) parser = qs.parse;
1104
1105   location = lolcation(location);
1106
1107   //
1108   // Extract protocol information before running the instructions.
1109   //
1110   extracted = extractProtocol(address || "");
1111   relative = extracted.protocol && extracted.slashes;
1112   url.slashes = extracted.slashes || relative && location.slashes;
1113   url.protocol = extracted.protocol || location.protocol || "";
1114   address = extracted.rest
1115
1116   //

```

```

1117 // When the authority component is absent the URL starts with a path
1118 // component.
1119 //
1120 if (!extracted.slashes) instructions[2] = [/(.*)/, 'pathname'];
1121
1122 for (; i < instructions.length; i++) {
1123   instruction = instructions[i];
1124   parse = instruction[0];
1125   key = instruction[1];
1126
1127   if (parse !== parse) {
1128     url[key] = address;
1129   } else if ('string' === typeof parse) {
1130     if (~index = address.indexOf(parse)) {
1131       if ('number' === typeof instruction[2]) {
1132         url[key] = address.slice(0, index);
1133         address = address.slice(index + instruction[2]);
1134       } else {
1135         url[key] = address.slice(index);
1136         address = address.slice(0, index);
1137       }
1138     }
1139   } else if (index = parse.exec(address)) {
1140     url[key] = index[1];
1141     address = address.slice(0, index.index);
1142   }
1143
1144   url[key] = url[key] || (
1145     relative && instruction[3] ? location[key] || '' : ''
1146   );
1147
1148   //
1149   // Hostname, host and protocol should be lowercased so they can be used to
1150   // create a proper `origin`.
1151   //
1152   if (instruction[4]) url[key] = url[key].toLowerCase();
1153 }
1154
1155 //
1156 // Also parse the supplied query string in to an object. If we're supplied
1157 // with a custom parser as function use that instead of the default build-in
1158 // parser.
1159 //
1160 if (parser) url.query = parser(url.query);
1161
1162 //
1163 // If the URL is relative, resolve the pathname against the base URL.
1164 //
1165 if (
1166   relative
1167   && location.slashes
1168   && url.pathname.charAt(0) !== '/'
1169   && (url.pathname !== '' || location.pathname !== '')
1170 ) {
1171   url.pathname = resolve(url.pathname, location.pathname);
1172 }
1173
1174 //
1175 // We should not add port numbers if they are already the default port number
1176 // for a given protocol. As the host also contains the port number we're going
1177 // override it with the hostname which contains no port number.
1178 //
1179 if (!required.url.port || url.protocol) {
1180   url.host = url.hostname;
1181   url.port = '';
1182 }
1183
1184 //
1185 // Parse down the `auth` for the username and password.
1186 //
1187 url.username = url.password = '';
1188 if (url.auth) {
1189   instruction = url.auth.split(':');
1190   url.username = instruction[0] || '';
1191   url.password = instruction[1] || '';
1192 }
1193
1194 url.origin = url.protocol && url.host && url.protocol !== 'file:'
1195   ? url.protocol + '//' + url.host
1196   : 'null';
1197
1198 //
1199 // The href is just the compiled result.
1200 //
1201 url.href = url.toString();
1202 }
1203
1204 /**
1205 * This is convenience method for changing properties in the URL instance to
1206 * insure that they all propagate correctly.
1207 *
1208 * @param {String} part      Property we need to adjust.
1209 * @param {Mixed} value      The newly assigned value.
1210 * @param {Boolean|Function} fn When setting the query, it will be the function
1211 *                               used to parse the query.
1212 *                               When setting the protocol, double slash will be
1213 *                               removed from the final url if it is true.
1214 * @returns {URL}
1215 * @api public
1216 */
1217 URL.prototype.set = function set(part, value, fn) {
1218   var url = this;
1219
1220   switch (part) {
1221     case 'query':
1222       if ('string' === typeof value && value.length) {
1223         value = (fn || qs.parse)(value);
1224       }
1225
1226       url[part] = value;
1227       break;
1228     case 'port':
1229       url[part] = value;
1230   }
1231 }

```

```

1232     if (!required(value, url.protocol)) {
1233         url.host = url.hostname;
1234         url.port = "";
1235     } else if (value) {
1236         url.host = url.hostname + ':' + value;
1237     }
1238
1239     break;
1240
1241     case 'hostname':
1242         url.port = value;
1243
1244         if (url.port) value += ':' + url.port;
1245         url.host = value;
1246         break;
1247
1248     case 'host':
1249         url.port = value;
1250
1251         if (/^(\d+$)/.test(value)) {
1252             value = value.split(':');
1253             url.port = value.pop();
1254             url.hostname = value.join(':');
1255         } else {
1256             url.hostname = value;
1257             url.port = "";
1258         }
1259
1260         break;
1261
1262     case 'protocol':
1263         url.protocol = value.toLowerCase();
1264         url.slashes = 'fn';
1265         break;
1266
1267     case 'pathname':
1268         url.pathname = value.length && value.charAt(0) !== '/' ? '/' + value : value;
1269
1270         break;
1271
1272     default:
1273         url.port = value;
1274     }
1275
1276     for (var i = 0; i < rules.length; i++) {
1277         var ins = rules[i];
1278
1279         if (ins[4]) url[ins[1]] = url[ins[1]].toLowerCase();
1280     }
1281
1282     url.origin = url.protocol && url.host && url.protocol !== 'file:'
1283         ? url.protocol + '//' + url.host
1284         : 'null';
1285
1286     url.href = url.toString();
1287
1288     return url;
1289 };
1290
1291 /**
1292  * Transform the properties back in to a valid and full URL string.
1293  *
1294  * @param {Function} stringify Optional query stringify function.
1295  * @returns {String}
1296  * @api public
1297  */
1298 URL.prototype.toString = function toString(stringify) {
1299     if (!stringify || 'function' !== typeof stringify) stringify = qs.stringify;
1300
1301     var query
1302         , url = this
1303         , protocol = url.protocol;
1304
1305     if (protocol && protocol.charAt(protocol.length - 1) !== ':') protocol += ':';
1306
1307     var result = protocol + (url.slashes ? '//' : '');
1308
1309     if (url.username) {
1310         result += url.username;
1311         if (url.password) result += ':' + url.password;
1312         result += '@';
1313     }
1314
1315     result += url.host + url.pathname;
1316
1317     query = 'object' === typeof url.query ? stringify(url.query) : url.query;
1318     if (query) result += '?' + query.charAt(0) ? '?' + query : query;
1319
1320     if (url.hash) result += url.hash;
1321
1322     return result;
1323 };
1324
1325 //
1326 // Expose the URL parser and some additional properties that might be useful for
1327 // others or testing.
1328 //
1329 URL.extractProtocol = extractProtocol;
1330 URL.location = lolcation;
1331 URL.qs = qs;
1332
1333 module.exports = URL;
1334
1335 }, {"/lolcation":20,"querystringify":17,"requires-port":18}],20:[function(require,module,exports){
1336 (function (global){
1337 'use strict';
1338
1339 var slashes = /^[A-Za-z][A-Za-z0-9+-.]*:\/?/;
1340
1341 /**
1342  * These properties should not be copied or inherited from. This is only needed
1343  * for all non blob URL's as a blob URL does not include a hash, only the
1344  * origin.
1345  *
1346  * @type {Object}

```

```

1347 * @private
1348 */
1349 var ignore = { hash: 1, query: 1 }
1350 , URL;
1351
1352 /**
1353 * The location object differs when your code is loaded through a normal page,
1354 * Worker or through a worker using a blob. And with the blobble begins the
1355 * trouble as the location object will contain the URL of the blob, not the
1356 * location of the page where our code is loaded in. The actual origin is
1357 * encoded in the `pathname` so we can thankfully generate a good "default"
1358 * location from it so we can generate proper relative URL's again.
1359 *
1360 * @param {Object|String} loc Optional default location object.
1361 * @returns {Object} lolocation object.
1362 * @api public
1363 */
1364 module.exports = function lolocation(loc) {
1365   loc = loc || global.location || {};
1366   URL = URL || require('url');
1367
1368   var finaldestination = {}
1369     , type = typeof loc
1370     , key;
1371
1372   if ('blob:' === loc.protocol) {
1373     finaldestination = new URL(unescape(loc.pathname), {});
1374   } else if ('string' === type) {
1375     finaldestination = new URL(loc, {});
1376     for (key in ignore) delete finaldestination[key];
1377   } else if ('object' === type) {
1378     for (key in loc) {
1379       if (key in ignore) continue;
1380       finaldestination[key] = loc[key];
1381     }
1382
1383     if (finaldestination.slashes === undefined) {
1384       finaldestination.slashes = slashes.test(loc.href);
1385     }
1386   }
1387
1388   return finaldestination;
1389 };
1390
1391 }) call(this, typeof global !== "undefined" ? global : typeof self !== "undefined" ? self : typeof window !== "undefined" ? window : {});
1392
1393 }, {"/":19}],21:(function(require,module,exports){
1394 (function (root, factory) {
1395   if (typeof exports === 'object') {
1396     module.exports = factory();
1397   } else if (typeof define === 'function' && define.amd) {
1398     define([], factory);
1399   } else {
1400     root.urltemplate = factory();
1401   }
1402 })(this, function () {
1403   /**
1404    * @constructor
1405    */
1406   function UriTemplate() {}
1407
1408   /**
1409    * @private
1410    * @param {string} str
1411    * @return {string}
1412    */
1413   UriTemplate.prototype.encodeReserved = function (str) {
1414     return str.split(/%[0-9A-Fa-f]{2}/g).map(function (part) {
1415       if (!/%[0-9A-Fa-f]/.test(part)) {
1416         part = encodeURIComponent(part).replace(/%5B/g, '[').replace(/%5D/g, ']');
1417       }
1418       return part;
1419     }).join("");
1420   };
1421
1422   /**
1423    * @private
1424    * @param {string} str
1425    * @return {string}
1426    */
1427   UriTemplate.prototype.encodeUnreserved = function (str) {
1428     return encodeURIComponent(str).replace(/!|'|"|/g, function (c) {
1429       return '%' + c.charCodeAt(0).toString(16).toUpperCase();
1430     });
1431   };
1432
1433   /**
1434    * @private
1435    * @param {string} operator
1436    * @param {string} value
1437    * @param {string} key
1438    * @return {string}
1439    */
1440   UriTemplate.prototype.encodeValue = function (operator, value, key) {
1441     value = (operator === '+' || operator === '#') ? this.encodeReserved(value) : this.encodeUnreserved(value);
1442
1443     if (key) {
1444       return this.encodeUnreserved(key) + '=' + value;
1445     } else {
1446       return value;
1447     }
1448   };
1449
1450   /**
1451    * @private
1452    * @param {*} value
1453    * @return {boolean}
1454    */
1455   UriTemplate.prototype.isDefined = function (value) {
1456     return value !== undefined && value !== null;
1457   };
1458
1459   /**
1460    * @private
1461

```

```

1462 * @param {string}
1463 * @return {boolean}
1464 */
1465 UriTemplate.prototype.isKeyOperator = function (operator) {
1466     return operator === ';' || operator === '&' || operator === '?';
1467 };
1468
1469 /**
1470 * @private
1471 * @param {Object} context
1472 * @param {string} operator
1473 * @param {string} key
1474 * @param {string} modifier
1475 */
1476 UriTemplate.prototype.getValues = function (context, operator, key, modifier) {
1477     var value = context[key],
1478         result = [];
1479
1480     if (this.isDefined(value) && value !== "") {
1481         if (typeof value === 'string' || typeof value === 'number' || typeof value === 'boolean') {
1482             value = value.toString();
1483
1484             if (modifier && modifier !== '*') {
1485                 value = value.substring(0, parseInt(modifier, 10));
1486             }
1487
1488             result.push(this.encodeValue(operator, value, this.isKeyOperator(operator) ? key : null));
1489         } else {
1490             if (modifier === '*') {
1491                 if (Array.isArray(value)) {
1492                     value.filter(this.isDefined).forEach(function (value) {
1493                         result.push(this.encodeValue(operator, value, this.isKeyOperator(operator) ? key : null));
1494                     }, this);
1495                 } else {
1496                     Object.keys(value).forEach(function (k) {
1497                         if (this.isDefined(value[k])) {
1498                             result.push(this.encodeValue(operator, value[k], k));
1499                         }
1500                     }, this);
1501                 }
1502             } else {
1503                 var tmp = [];
1504
1505                 if (Array.isArray(value)) {
1506                     value.filter(this.isDefined).forEach(function (value) {
1507                         tmp.push(this.encodeValue(operator, value));
1508                     }, this);
1509                 } else {
1510                     Object.keys(value).forEach(function (k) {
1511                         if (this.isDefined(value[k])) {
1512                             tmp.push(this.encodeUnreserved(k));
1513                             tmp.push(this.encodeValue(operator, value[k].toString()));
1514                         }
1515                     }, this);
1516                 }
1517
1518                 if (this.isKeyOperator(operator)) {
1519                     result.push(this.encodeUnreserved(key) + '=' + tmp.join(','));
1520                 } else if (tmp.length !== 0) {
1521                     result.push(tmp.join(','));
1522                 }
1523             }
1524         }
1525     } else {
1526         if (operator === ';' || operator === '&' || operator === '?') {
1527             if (this.isDefined(value)) {
1528                 result.push(this.encodeUnreserved(key));
1529             }
1530         } else if (value === "" && (operator === '&' || operator === '?')) {
1531             result.push(this.encodeUnreserved(key) + '=');
1532         } else if (value === "") {
1533             result.push("");
1534         }
1535     }
1536     return result;
1537 };
1538
1539 /**
1540 * @param {string} template
1541 * @return {function(Object):string}
1542 */
1543 UriTemplate.prototype.parse = function (template) {
1544     var that = this;
1545     var operators = ['+', '#', '.', '/', ':', '?', '&'];
1546
1547     return {
1548         expand: function (context) {
1549             return template.replace(/\{([^\{\}]+\})\}/g, function (_, expression, literal) {
1550                 if (expression) {
1551                     var operator = null,
1552                         values = [];
1553
1554                     if (operators.indexOf(expression.charAt(0)) !== -1) {
1555                         operator = expression.charAt(0);
1556                         expression = expression.substr(1);
1557                     }
1558
1559                     expression.split(/,/g).forEach(function (variable) {
1560                         var tmp = /([^\:\*\?]{1,}(\d+)?(\*)?)?/.exec(variable);
1561                         values.push.apply(values, that.getValues(context, operator, tmp[1], tmp[2] || tmp[3]));
1562                     });
1563
1564                     if (operator && operator !== '+') {
1565                         var separator = ',';
1566
1567                         if (operator === '?') {
1568                             separator = '&';
1569                         } else if (operator !== '#') {
1570                             separator = operator;
1571                         }
1572
1573                         return (values.length !== 0 ? operator : "") + values.join(separator);
1574                     } else {
1575                         return values.join(',');
1576                     }
1577                 }
1578             });
1579         }
1580     };

```

```

1577         return that.encodeReservedLiteral();
1578     }
1579 });
1580 }
1581 };
1582 };
1583
1584 return new UriTemplate();
1585 });
1586
1587 }, {}), 22, (function(require, module, exports) {
1588     (function(self) {
1589         'use strict';
1590
1591         if (self.fetch) {
1592             return
1593         }
1594
1595         var support = {
1596             searchParams: 'URLSearchParams' in self,
1597             iterable: 'Symbol' in self && 'iterator' in Symbol,
1598             blob: 'FileReader' in self && 'Blob' in self && (function() {
1599                 try {
1600                     new Blob()
1601                     return true
1602                 } catch (e) {
1603                     return false
1604                 }
1605             })(),
1606             formData: 'FormData' in self,
1607             arrayBuffer: 'ArrayBuffer' in self
1608         }
1609
1610         if (support.arrayBuffer) {
1611             var viewClasses = [
1612                 '[object Int8Array]',
1613                 '[object Uint8Array]',
1614                 '[object Uint8ClampedArray]',
1615                 '[object Int16Array]',
1616                 '[object Uint16Array]',
1617                 '[object Int32Array]',
1618                 '[object Uint32Array]',
1619                 '[object Float32Array]',
1620                 '[object Float64Array]'
1621             ]
1622
1623             var isDataView = function(obj) {
1624                 return obj && DataView.prototype.isPrototypeOf(obj)
1625             }
1626
1627             var isArrayBufferView = ArrayBuffer.isView || function(obj) {
1628                 return obj && viewClasses.indexOf(Object.prototype.toString.call(obj)) > -1
1629             }
1630         }
1631
1632         function normalizeName(name) {
1633             if (typeof name !== 'string') {
1634                 name = String(name)
1635             }
1636             if (/^[^a-z0-9_\#$%&*+.\^_-]~$/i.test(name)) {
1637                 throw new TypeError('Invalid character in header field name')
1638             }
1639             return name.toLowerCase()
1640         }
1641
1642         function normalizeValue(value) {
1643             if (typeof value !== 'string') {
1644                 value = String(value)
1645             }
1646             return value
1647         }
1648
1649         // Build a destructive iterator for the value list
1650         function iteratorFor(items) {
1651             var iterator = {
1652                 next: function() {
1653                     var value = items.shift()
1654                     return {done: value === undefined, value: value}
1655                 }
1656             }
1657
1658             if (support.iterable) {
1659                 iterator[Symbol.iterator] = function() {
1660                     return iterator
1661                 }
1662             }
1663
1664             return iterator
1665         }
1666
1667         function Headers(headers) {
1668             this.map = {}
1669
1670             if (headers instanceof Headers) {
1671                 headers.forEach(function(value, name) {
1672                     this.append(name, value)
1673                 }, this)
1674             } else if (headers) {
1675                 Object.getOwnPropertyNames(headers).forEach(function(name) {
1676                     this.append(name, headers[name])
1677                 }, this)
1678             }
1679         }
1680
1681         Headers.prototype.append = function(name, value) {
1682             name = normalizeName(name)
1683             value = normalizeValue(value)
1684             var oldValue = this.map[name]
1685             this.map[name] = oldValue + ', ' + value + value
1686         }
1687
1688         Headers.prototype['delete'] = function(name) {
1689             delete this.map[normalizeName(name)]
1690         }
1691     })

```

```

1692
1693 Headers.prototype.get = function(name) {
1694   name = normalizeName(name)
1695   return this.has(name) ? this.map[name] : null
1696 }
1697
1698 Headers.prototype.has = function(name) {
1699   return this.map.hasOwnProperty(normalizeName(name))
1700 }
1701
1702 Headers.prototype.set = function(name, value) {
1703   this.map[normalizeName(name)] = normalizeValue(value)
1704 }
1705
1706 Headers.prototype.forEach = function(callback, thisArg) {
1707   for (var name in this.map) {
1708     if (this.map.hasOwnProperty(name)) {
1709       callback.call(thisArg, this.map[name], name, this)
1710     }
1711   }
1712 }
1713
1714 Headers.prototype.keys = function() {
1715   var items = []
1716   this.forEach(function(value, name) { items.push(name) })
1717   return iteratorFor(items)
1718 }
1719
1720 Headers.prototype.values = function() {
1721   var items = []
1722   this.forEach(function(value) { items.push(value) })
1723   return iteratorFor(items)
1724 }
1725
1726 Headers.prototype.entries = function() {
1727   var items = []
1728   this.forEach(function(value, name) { items.push([name, value]) })
1729   return iteratorFor(items)
1730 }
1731
1732 if (supportIterable) {
1733   Headers.prototype[Symbol.iterator] = Headers.prototype.entries
1734 }
1735
1736 function consumedBody() {
1737   if (body.bodyUsed) {
1738     return Promise.reject(new TypeError('Already read'))
1739   }
1740   body.bodyUsed = true
1741 }
1742
1743 function fileReaderReady(reader) {
1744   return new Promise(function(resolve, reject) {
1745     reader.onload = function() {
1746       resolve(reader.result)
1747     }
1748     reader.onerror = function() {
1749       reject(reader.error)
1750     }
1751   })
1752 }
1753
1754 function readBlobAsArrayBuffer(blob) {
1755   var reader = new FileReader()
1756   var promise = fileReaderReady(reader)
1757   reader.readAsArrayBuffer(blob)
1758   return promise
1759 }
1760
1761 function readBlobAsText(blob) {
1762   var reader = new FileReader()
1763   var promise = fileReaderReady(reader)
1764   reader.readAsText(blob)
1765   return promise
1766 }
1767
1768 function bufferClone(buf) {
1769   if (buf.slice) {
1770     return buf.slice(0)
1771   } else {
1772     var view = new Uint8Array(buf.byteLength)
1773     view.set(new Uint8Array(buf))
1774     return view.buffer
1775   }
1776 }
1777
1778 function Body() {
1779   this.bodyUsed = false
1780
1781   this._initBody = function(body) {
1782     this._bodyInit = body
1783     if (!body) {
1784       this._bodyText = ''
1785     } else if (typeof body === 'string') {
1786       this._bodyText = body
1787     } else if (supportBlob && Blob.prototype.isPrototypeOf(body)) {
1788       this._bodyBlob = body
1789     } else if (supportFormData && FormData.prototype.isPrototypeOf(body)) {
1790       this._bodyFormData = body
1791     } else if (supportSearchParams && URLSearchParams.prototype.isPrototypeOf(body)) {
1792       this._bodyText = body.toString()
1793     } else if (supportArrayBuffer && supportBlob && isDataView(body)) {
1794       this._bodyArrayBuffer = bufferClone(body.buffer)
1795       // IE 10-11 can't handle a DataView body.
1796       this._bodyInit = new Blob([this._bodyArrayBuffer])
1797     } else if (supportArrayBuffer && (ArrayBuffer.prototype.isPrototypeOf(body) || isArrayBufferView(body))) {
1798       this._bodyArrayBuffer = bufferClone(body)
1799     } else {
1800       throw new Error('unsupported BodyInit type')
1801     }
1802   }
1803
1804   if (!this.headers.get('content-type')) {
1805     if (typeof body === 'string') {
1806       this.headers.set('content-type', 'text/plain;charset=UTF-8')
1807     } else if (this._bodyBlob && this._bodyBlob.type) {

```

```

1807     this.headers.set('content-type', this._bodyBlob.type)
1808   } else if (support.searchParams && URLSearchParams.prototype.isPrototypeOf(body)) {
1809     this.headers.set('content-type', 'application/x-www-form-urlencoded;charset=UTF-8')
1810   }
1811 }
1812 }
1813
1814 if (support.blob) {
1815   this.blob = function() {
1816     var rejected = consumed(this)
1817     if (rejected) {
1818       return rejected
1819     }
1820
1821     if (this._bodyBlob) {
1822       return Promise.resolve(this._bodyBlob)
1823     } else if (this._bodyArrayBuffer) {
1824       return Promise.resolve(new Blob([this._bodyArrayBuffer]))
1825     } else if (this._bodyFormData) {
1826       throw new Error('could not read FormData body as blob')
1827     } else {
1828       return Promise.resolve(new Blob([this._bodyText]))
1829     }
1830   }
1831 }
1832
1833 this.text = function() {
1834   var rejected = consumed(this)
1835   if (rejected) {
1836     return rejected
1837   }
1838
1839   if (this._bodyBlob) {
1840     return readBlobAsText(this._bodyBlob)
1841   } else if (this._bodyArrayBuffer) {
1842     var view = new Uint8Array(this._bodyArrayBuffer)
1843     var str = String.fromCharCode.apply(null, view)
1844     return Promise.resolve(str)
1845   } else if (this._bodyFormData) {
1846     throw new Error('could not read FormData body as text')
1847   } else {
1848     return Promise.resolve(this._bodyText)
1849   }
1850 }
1851
1852 if (support.arrayBuffer) {
1853   this.arrayBuffer = function() {
1854     if (this._bodyArrayBuffer) {
1855       return consumed(this) || Promise.resolve(this._bodyArrayBuffer)
1856     } else {
1857       return this.blob().then(readBlobAsArrayBuffer)
1858     }
1859   }
1860 }
1861
1862 if (support.formData) {
1863   this.formData = function() {
1864     return this.text().then(decode)
1865   }
1866 }
1867
1868 this.json = function() {
1869   return this.text().then(JSON.parse)
1870 }
1871
1872 return this
1873 }
1874
1875 // HTTP methods whose capitalization should be normalized
1876 var methods = ['DELETE', 'GET', 'HEAD', 'OPTIONS', 'POST', 'PUT']
1877
1878 function normalizeMethod(method) {
1879   var upcased = method.toUpperCase()
1880   return (methods.indexOf(upcased) > -1) ? upcased : method
1881 }
1882
1883 function Request(input, options) {
1884   options = options || {}
1885   var body = options.body
1886
1887   if (typeof input === 'string') {
1888     this.url = input
1889   } else {
1890     if (input.bodyUsed) {
1891       throw new TypeError('Already read')
1892     }
1893     this.url = input.url
1894     this.credentials = input.credentials
1895     if (!options.headers) {
1896       this.headers = new Headers(input.headers)
1897     }
1898     this.method = input.method
1899     this.mode = input.mode
1900     if (!body && input._bodyInit != null) {
1901       body = input._bodyInit
1902       input.bodyUsed = true
1903     }
1904   }
1905
1906   this.credentials = options.credentials || this.credentials || 'omit'
1907   if (options.headers || !this.headers) {
1908     this.headers = new Headers(options.headers)
1909   }
1910   this.method = normalizeMethod(options.method || this.method || 'GET')
1911   this.mode = options.mode || this.mode || null
1912   this.referrer = null
1913
1914   if ((this.method === 'GET' || this.method === 'HEAD') && body) {
1915     throw new TypeError('Body not allowed for GET or HEAD requests')
1916   }
1917   this._initBody(body)
1918 }
1919
1920 Request.prototype.clone = function() {
1921   return new Request(this, { body: this._bodyInit })

```



```

1922 }
1923
1924 function decodeBody() {
1925   var form = new FormData()
1926   body.trim().split('&').forEach(function(bytes) {
1927     if (bytes) {
1928       var split = bytes.split('=')
1929       var name = split.shift().replace(/\+/g, ' ')
1930       var value = split.join('=').replace(/\+/g, ' ')
1931       form.append(decodeURIComponent(name), decodeURIComponent(value))
1932     }
1933   })
1934   return form
1935 }
1936
1937 function parseHeaders(rawHeaders) {
1938   var headers = new Headers()
1939   rawHeaders.split('\r\n').forEach(function(line) {
1940     var parts = line.split(':')
1941     var key = parts.shift().trim()
1942     if (key) {
1943       var value = parts.join(':').trim()
1944       headers.append(key, value)
1945     }
1946   })
1947   return headers
1948 }
1949
1950 Body.call(Request.prototype)
1951
1952 function Response(bodyInit, options) {
1953   if (!options) {
1954     options = {}
1955   }
1956
1957   this.type = 'default'
1958   this.status = 'status' in options ? options.status : 200
1959   this.ok = this.status >= 200 && this.status < 300
1960   this.statusText = 'statusText' in options ? options.statusText : 'OK'
1961   this.headers = new Headers(options.headers)
1962   this.url = options.url || ''
1963   this._initBody(bodyInit)
1964 }
1965
1966 Body.call(Response.prototype)
1967
1968 Response.prototype.clone = function() {
1969   return new Response(this._bodyInit, {
1970     status: this.status,
1971     statusText: this.statusText,
1972     headers: new Headers(this.headers),
1973     url: this.url
1974   })
1975 }
1976
1977 Response.error = function() {
1978   var response = new Response(null, {status: 0, statusText: ''})
1979   response.type = 'error'
1980   return response
1981 }
1982
1983 var redirectStatuses = [301, 302, 303, 307, 308]
1984
1985 Response.redirect = function(url, status) {
1986   if (redirectStatuses.indexOf(status) === -1) {
1987     throw new RangeError('Invalid status code')
1988   }
1989
1990   return new Response(null, {status: status, headers: {location: url}})
1991 }
1992
1993 self.Headers = Headers
1994 self.Request = Request
1995 self.Response = Response
1996
1997 self.fetch = function(input, init) {
1998   return new Promise(function(resolve, reject) {
1999     var request = new Request(input, init)
2000     var xhr = new XMLHttpRequest()
2001
2002     xhr.onload = function() {
2003       var options = {
2004         status: xhr.status,
2005         statusText: xhr.statusText,
2006         headers: parseHeaders(xhr.getAllResponseHeaders()) || ''
2007       }
2008       options.url = 'responseURL' in xhr ? xhr.responseURL : options.headers.get('X-Request-URL')
2009       var body = 'response' in xhr ? xhr.response : xhr.responseText
2010       resolve(new Response(body, options))
2011     }
2012
2013     xhr.onerror = function() {
2014       reject(new TypeError('Network request failed'))
2015     }
2016
2017     xhr.ontimeout = function() {
2018       reject(new TypeError('Network request failed'))
2019     }
2020
2021     xhr.open(request.method, request.url, true)
2022
2023     if (request.credentials === 'include') {
2024       xhr.withCredentials = true
2025     }
2026
2027     if (!('responseType' in xhr) && supportBlob) {
2028       xhr.responseType = 'blob'
2029     }
2030
2031     request.headers.forEach(function(value, name) {
2032       xhr.setRequestHeader(name, value)
2033     })
2034
2035     xhr.send(typeof request._bodyInit === 'undefined' ? null : request._bodyInit)
2036   })

```

```

2037 }
2038 self.fetch.polyfill = true
2039 })(typeof self !== 'undefined' ? self : this);
2040
2041 },{}),{}),[12]](12)
2042 });
2043

```

**File: ./static/rest\_framework/img/glyphicons-halflings-white.png**

[binary]

**File: ./static/rest\_framework/img/grid.png**

[binary]

**File: ./static/rest\_framework/img/glyphicons-halflings.png**

[binary]

**File: ./static/admin/css/rtl.css**

```

1  /* GLOBAL */
2
3  th {
4      text-align: right;
5  }
6
7  .module h2, .module caption {
8      text-align: right;
9  }
10
11  .module ul, .module ol {
12      margin-left: 0;
13      margin-right: 1.5em;
14  }
15
16  .viewlink, .addlink, .changelink {
17      padding-left: 0;
18      padding-right: 16px;
19      background-position: 100% 1px;
20  }
21
22  .deletelink {
23      padding-left: 0;
24      padding-right: 16px;
25      background-position: 100% 1px;
26  }
27
28  .object-tools {
29      float: left;
30  }
31
32  thead th:first-child,
33  tfoot td:first-child {
34      border-left: none;
35  }
36
37  /* LAYOUT */
38
39  #user-tools {
40      right: auto;
41      left: 0;
42      text-align: left;
43  }
44
45  div.breadcrumbs {
46      text-align: right;
47  }
48
49  #content-main {
50      float: right;
51  }
52
53  #content-related {
54      float: left;
55      margin-left: -300px;
56      margin-right: auto;
57  }
58
59  .colMS {

```

```

60     margin-left: 300px;
61     margin-right: 0;
62 }
63
64 /* SORTABLE TABLES */
65
66 table thead th.sorted .sortoptions {
67     float: left;
68 }
69
70 thead th.sorted .text {
71     padding-right: 0;
72     padding-left: 42px;
73 }
74
75 /* dashboard styles */
76
77 .dashboard .module table td a {
78     padding-left: .6em;
79     padding-right: 16px;
80 }
81
82 /* changelists styles */
83
84 .change-list .filtered table {
85     border-left: none;
86     border-right: 0px none;
87 }
88
89 #changelist-filter {
90     border-left: none;
91     border-right: none;
92     margin-left: 0;
93     margin-right: 30px;
94 }
95
96 #changelist-filter li.selected {
97     border-left: none;
98     padding-left: 10px;
99     margin-left: 0;
100    border-right: 5px solid var(--hairline-color);
101    padding-right: 10px;
102    margin-right: -15px;
103 }
104
105 #changelist table tbody td:first-child, #changelist table tbody th:first-child {
106     border-right: none;
107     border-left: none;
108 }
109
110 /* FORMS */
111
112 .aligned label {
113     padding: 0 0 3px 1em;
114     float: right;
115 }
116
117 .submit-row {
118     text-align: left;
119 }
120
121 .submit-row p.deletelink-box {
122     float: right;
123 }
124
125 .submit-row input.default {
126     margin-left: 0;
127 }
128
129 .vDateField, .vTimeField {
130     margin-left: 2px;
131 }
132
133 .aligned .form-row input {
134     margin-left: 5px;
135 }
136
137 form .aligned p.help, form .aligned div.help {
138     clear: right;
139 }
140
141 form .aligned ul {
142     margin-right: 163px;
143     margin-left: 0;
144 }
145
146 form ul.inline li {
147     float: right;
148     padding-right: 0;
149     padding-left: 7px;
150 }
151
152 input[type=submit].default, .submit-row input.default {
153     float: left;
154 }
155
156 fieldset .fieldBox {
157     float: right;
158     margin-left: 20px;
159     margin-right: 0;
160 }
161
162 .errorlist li {
163     background-position: 100% 12px;
164     padding: 0;
165 }
166
167 .errornote {
168     background-position: 100% 12px;
169     padding: 10px 12px;
170 }
171
172 /* WIDGETS */
173
174 .calendarnav-previous {

```

```

175 top: 0;
176 left: auto;
177 right: 10px;
178 }
179
180 calendarnav-next {
181 top: 0;
182 right: auto;
183 left: 10px;
184 }
185
186 calendar caption, .calendarbox h2 {
187 text-align: center;
188 }
189
190 selector {
191 float: right;
192 }
193
194 selector.selector-filter {
195 text-align: right;
196 }
197
198 inline-deletelink {
199 float: left;
200 }
201
202 form .form-row p.datetime {
203 overflow: hidden;
204 }
205
206 related-widget-wrapper {
207 float: right;
208 }
209
210 /* MISC */
211
212 inline-related h2, .inline-group h2 {
213 text-align: right;
214 }
215
216 inline-related h3 span.delete {
217 padding-right: 20px;
218 padding-left: inherit;
219 left: 10px;
220 right: inherit;
221 float: left;
222 }
223
224 inline-related h3 span.delete label {
225 margin-left: inherit;
226 margin-right: 2px;
227 }

```

**File: ./static/admin/css/changelists.css**

```

1  /* CHANGELISTS */
2
3  #changelist {
4      display: flex;
5      align-items: flex-start;
6      justify-content: space-between;
7  }
8
9  #changelist .changelist-form-container {
10     flex: 1 1 auto;
11     min-width: 0;
12 }
13
14 #changelist table {
15     width: 100%;
16 }
17
18 .change-list .hiddenfields { display: none; }
19
20 .change-list .filtered table {
21     border-right: none;
22 }
23
24 .change-list .filtered {
25     min-height: 400px;
26 }
27
28 .change-list .filtered .results, .change-list .filtered .paginator,
29 .filtered #toolbar, .filtered div.xfull {
30     width: auto;
31 }
32
33 .change-list .filtered table tbody th {
34     padding-right: 1em;
35 }
36
37 #changelist-form .results {
38     overflow-x: auto;
39     width: 100%;
40 }
41
42 #changelist .toplinks {
43     border-bottom: 1px solid var(--hairline-color);
44 }
45
46 #changelist .paginator {
47     color: var(--body-quiet-color);
48     border-bottom: 1px solid var(--hairline-color);
49     background: var(--body-bg);
50     overflow: hidden;
51 }
52
53 /* CHANGELIST TABLES */
54
55 #changelist table thead th {

```

```

56 padding: 0;
57 white-space: nowrap;
58 vertical-align: middle;
59 }
60
61 #changelist table thead th.action-checkbox-column {
62 width: 1.5em;
63 text-align: center;
64 }
65
66 #changelist table tbody td.action-checkbox {
67 text-align: center;
68 }
69
70 #changelist table tfoot {
71 color: var(--body-quiet-color);
72 }
73
74 /* TOOLBAR */
75
76 #toolbar {
77 padding: 8px 10px;
78 margin-bottom: 15px;
79 border-top: 1px solid var(--hairline-color);
80 border-bottom: 1px solid var(--hairline-color);
81 background: var(--darkened-bg);
82 color: var(--body-quiet-color);
83 }
84
85 #toolbar form input {
86 border-radius: 4px;
87 font-size: 14px;
88 padding: 5px;
89 color: var(--body-fg);
90 }
91
92 #toolbar #searchbar {
93 height: 19px;
94 border: 1px solid var(--border-color);
95 padding: 2px 5px;
96 margin: 0;
97 vertical-align: top;
98 font-size: 13px;
99 max-width: 100%;
100 }
101
102 #toolbar #searchbar:focus {
103 border-color: var(--body-quiet-color);
104 }
105
106 #toolbar form input[type="submit"] {
107 border: 1px solid var(--border-color);
108 font-size: 13px;
109 padding: 4px 8px;
110 margin: 0;
111 vertical-align: middle;
112 background: var(--body-bg);
113 box-shadow: 0 -15px 20px -10px rgba(0, 0, 0, 0.15) inset;
114 cursor: pointer;
115 color: var(--body-fg);
116 }
117
118 #toolbar form input[type="submit"]:focus,
119 #toolbar form input[type="submit"]:hover {
120 border-color: var(--body-quiet-color);
121 }
122
123 #changelist-search img {
124 vertical-align: middle;
125 margin-right: 4px;
126 }
127
128 #changelist-search .help {
129 word-break: break-word;
130 }
131
132 /* FILTER COLUMN */
133
134 #changelist-filter {
135 flex: 0 0 240px;
136 order: 1;
137 background: var(--darkened-bg);
138 border-left: none;
139 margin: 0 0 30px;
140 }
141
142 #changelist-filter h2 {
143 font-size: 14px;
144 text-transform: uppercase;
145 letter-spacing: 0.5px;
146 padding: 5px 15px;
147 margin-bottom: 12px;
148 border-bottom: none;
149 }
150
151 #changelist-filter h3 {
152 font-weight: 400;
153 padding: 0 15px;
154 margin-bottom: 10px;
155 }
156
157 #changelist-filter ul {
158 margin: 5px 0;
159 padding: 0 15px 15px;
160 border-bottom: 1px solid var(--hairline-color);
161 }
162
163 #changelist-filter ul:last-child {
164 border-bottom: none;
165 }
166
167 #changelist-filter li {
168 list-style-type: none;
169 margin-left: 0;
170 padding-left: 0;

```

```

171 }
172
173 #changelist-filter a {
174   display: block;
175   color: var(--body-quiet-color);
176   text-overflow: ellipsis;
177   overflow-x: hidden;
178 }
179
180 #changelist-filter li.selected {
181   border-left: 5px solid var(--hairline-color);
182   padding-left: 10px;
183   margin-left: -15px;
184 }
185
186 #changelist-filter li.selected a {
187   color: var(--link-selected-fg);
188 }
189
190 #changelist-filter a:focus, #changelist-filter a:hover,
191 #changelist-filter li.selected a:focus,
192 #changelist-filter li.selected a:hover {
193   color: var(--link-hover-color);
194 }
195
196 #changelist-filter #changelist-filter-clear a {
197   font-size: 13px;
198   padding-bottom: 10px;
199   border-bottom: 1px solid var(--hairline-color);
200 }
201
202 /* DATE DRILLDOWN */
203
204 .change-list ul.toplinks {
205   display: block;
206   float: left;
207   padding: 0;
208   margin: 0;
209   width: 100%;
210 }
211
212 .change-list ul.toplinks li {
213   padding: 3px 6px;
214   font-weight: bold;
215   list-style-type: none;
216   display: inline-block;
217 }
218
219 .change-list ul.toplinks .date-back a {
220   color: var(--body-quiet-color);
221 }
222
223 .change-list ul.toplinks .date-back a:focus,
224 .change-list ul.toplinks .date-back a:hover {
225   color: var(--link-hover-color);
226 }
227
228 /* PAGINATOR */
229
230 .paginator {
231   font-size: 13px;
232   padding-top: 10px;
233   padding-bottom: 10px;
234   line-height: 22px;
235   margin: 0;
236   border-top: 1px solid var(--hairline-color);
237   width: 100%;
238 }
239
240 .paginator a:link, .paginator a:visited {
241   padding: 2px 6px;
242   background: var(--button-bg);
243   text-decoration: none;
244   color: var(--button-fg);
245 }
246
247 .paginator a.showall {
248   border: none;
249   background: none;
250   color: var(--link-fg);
251 }
252
253 .paginator a.showall:focus, .paginator a.showall:hover {
254   background: none;
255   color: var(--link-hover-color);
256 }
257
258 .paginator .end {
259   margin-right: 6px;
260 }
261
262 .paginator .this-page {
263   padding: 2px 6px;
264   font-weight: bold;
265   font-size: 13px;
266   vertical-align: top;
267 }
268
269 .paginator a:focus, .paginator a:hover {
270   color: white;
271   background: var(--link-hover-color);
272 }
273
274 /* ACTIONS */
275
276 .filtered .actions {
277   border-right: none;
278 }
279
280 #changelist table input {
281   margin: 0;
282   vertical-align: baseline;
283 }
284
285 #changelist table tbody tr.selected {

```

```

286 background-color: var(--selected-row);
287 }
288
289 #changelist .actions {
290 padding: 10px;
291 background: var(--body-bg);
292 border-top: none;
293 border-bottom: none;
294 line-height: 24px;
295 color: var(--body-quiet-color);
296 width: 100%;
297 }
298
299 #changelist .actions.selected { /* XXXX Probably unused? */
300 background: var(--body-bg);
301 border-top: 1px solid var(--body-bg);
302 border-bottom: 1px solid #edecdc;
303 }
304
305 #changelist .actions span.all,
306 #changelist .actions span.action-counter,
307 #changelist .actions span.clear,
308 #changelist .actions span.question {
309 font-size: 13px;
310 margin: 0 0.5em;
311 }
312
313 #changelist .actions:last-child {
314 border-bottom: none;
315 }
316
317 #changelist .actions select {
318 vertical-align: top;
319 height: 24px;
320 color: var(--body-fg);
321 border: 1px solid var(--border-color);
322 border-radius: 4px;
323 font-size: 14px;
324 padding: 0 0 0 4px;
325 margin: 0;
326 margin-left: 10px;
327 }
328
329 #changelist .actions select:focus {
330 border-color: var(--body-quiet-color);
331 }
332
333 #changelist .actions label {
334 display: inline-block;
335 vertical-align: middle;
336 font-size: 13px;
337 }
338
339 #changelist .actions .button {
340 font-size: 13px;
341 border: 1px solid var(--border-color);
342 border-radius: 4px;
343 background: var(--body-bg);
344 box-shadow: 0 -15px 20px -10px rgba(0, 0, 0, 0.15) inset;
345 cursor: pointer;
346 height: 24px;
347 line-height: 1;
348 padding: 4px 8px;
349 margin: 0;
350 color: var(--body-fg);
351 }
352
353 #changelist .actions .button:focus, #changelist .actions .button:hover {
354 border-color: var(--body-quiet-color);
355 }

```

**File: ./static/admin/css/widgets.css**

```

1  /* SELECTOR (FILTER INTERFACE) */
2
3  .selector {
4      width: 800px;
5      float: left;
6  }
7
8  .selector select {
9      width: 380px;
10     height: 17.2em;
11 }
12
13 .selector-available, .selector-chosen {
14     float: left;
15     width: 380px;
16     text-align: center;
17     margin-bottom: 5px;
18 }
19
20 .selector-chosen select {
21     border-top: none;
22 }
23
24 .selector-available h2, .selector-chosen h2 {
25     border: 1px solid var(--border-color);
26     border-radius: 4px 4px 0 0;
27 }
28
29 .selector-chosen h2 {
30     background: var(--primary);
31     color: var(--header-link-color);
32 }
33
34 .selector .selector-available h2 {
35     background: var(--darkened-bg);

```

```

36     color: var(--body-quiet-color);
37 }
38
39 .selector_selector-filter {
40     border: 1px solid var(--border-color);
41     border-width: 0 1px;
42     padding: 8px;
43     color: var(--body-quiet-color);
44     font-size: 10px;
45     margin: 0;
46     text-align: left;
47 }
48
49 .selector_selector-filter label,
50 .inline-group .aligned .selector_selector-filter label {
51     float: left;
52     margin: 7px 0 0;
53     width: 18px;
54     height: 18px;
55     padding: 0;
56     overflow: hidden;
57     line-height: 1;
58 }
59
60 .selector_selector-available input {
61     width: 320px;
62     margin-left: 8px;
63 }
64
65 .selector ul.selector-chooser {
66     float: left;
67     width: 22px;
68     background-color: var(--selected-bg);
69     border-radius: 10px;
70     margin: 10em 5px 0 5px;
71     padding: 0;
72 }
73
74 .selector-chooser li {
75     margin: 0;
76     padding: 3px;
77     list-style-type: none;
78 }
79
80 .selector select {
81     padding: 0 10px;
82     margin: 0 0 10px;
83     border-radius: 0 0 4px 4px;
84 }
85
86 .selector-add, .selector-remove {
87     width: 16px;
88     height: 16px;
89     display: block;
90     text-indent: -3000px;
91     overflow: hidden;
92     cursor: default;
93     opacity: 0.55;
94 }
95
96 .active.selector-add, .active.selector-remove {
97     opacity: 1;
98 }
99
100 .active.selector-add:hover, .active.selector-remove:hover {
101     cursor: pointer;
102 }
103
104 .selector-add {
105     background: url(../img/selector-icons.svg) 0 -96px no-repeat;
106 }
107
108 .active.selector-add:focus, .active.selector-add:hover {
109     background-position: 0 -112px;
110 }
111
112 .selector-remove {
113     background: url(../img/selector-icons.svg) 0 -64px no-repeat;
114 }
115
116 .active.selector-remove:focus, .active.selector-remove:hover {
117     background-position: 0 -80px;
118 }
119
120 a.selector-chooseall, a.selector-clearall {
121     display: inline-block;
122     height: 16px;
123     text-align: left;
124     margin: 1px auto 3px;
125     overflow: hidden;
126     font-weight: bold;
127     line-height: 16px;
128     color: var(--body-quiet-color);
129     text-decoration: none;
130     opacity: 0.55;
131 }
132
133 a.active.selector-chooseall:focus, a.active.selector-clearall:focus,
134 a.active.selector-chooseall:hover, a.active.selector-clearall:hover {
135     color: var(--link-fg);
136 }
137
138 a.active.selector-chooseall, a.active.selector-clearall {
139     opacity: 1;
140 }
141
142 a.active.selector-chooseall:hover, a.active.selector-clearall:hover {
143     cursor: pointer;
144 }
145
146 a.selector-chooseall {
147     padding: 0 18px 0 0;
148     background: url(../img/selector-icons.svg) right -160px no-repeat;
149     cursor: default;
150 }

```



```

151
152 a.active.selector-chooseall:focus, a.active.selector-chooseall:hover {
153     background-position: 100% -176px;
154 }
155
156 a.selector-clearall {
157     padding: 0 0 0 18px;
158     background: url(../img/selector-icons.svg) 0 -128px no-repeat;
159     cursor: default;
160 }
161
162 a.active.selector-clearall:focus, a.active.selector-clearall:hover {
163     background-position: 0 -144px;
164 }
165
166 /* STACKED SELECTORS */
167
168 .stacked {
169     float: left;
170     width: 490px;
171 }
172
173 .stacked select {
174     width: 480px;
175     height: 10.1em;
176 }
177
178 .stacked .selector-available, .stacked .selector-chosen {
179     width: 480px;
180 }
181
182 .stacked .selector-available {
183     margin-bottom: 0;
184 }
185
186 .stacked .selector-available input {
187     width: 422px;
188 }
189
190 .stacked ul.selector-chooser {
191     height: 22px;
192     width: 50px;
193     margin: 0 0 10px 40%;
194     background-color: #eee;
195     border-radius: 10px;
196 }
197
198 .stacked .selector-chooser li {
199     float: left;
200     padding: 3px 3px 3px 5px;
201 }
202
203 .stacked .selector-chooseall, .stacked .selector-clearall {
204     display: none;
205 }
206
207 .stacked .selector-add {
208     background: url(../img/selector-icons.svg) 0 -32px no-repeat;
209     cursor: default;
210 }
211
212 .stacked .active.selector-add {
213     background-position: 0 -32px;
214     cursor: pointer;
215 }
216
217 .stacked .active.selector-add:focus, .stacked .active.selector-add:hover {
218     background-position: 0 -48px;
219     cursor: pointer;
220 }
221
222 .stacked .selector-remove {
223     background: url(../img/selector-icons.svg) 0 0 no-repeat;
224     cursor: default;
225 }
226
227 .stacked .active.selector-remove {
228     background-position: 0 0px;
229     cursor: pointer;
230 }
231
232 .stacked .active.selector-remove:focus, .stacked .active.selector-remove:hover {
233     background-position: 0 -16px;
234     cursor: pointer;
235 }
236
237 .selector .help-icon {
238     background: url(../img/icon-unknown.svg) 0 0 no-repeat;
239     display: inline-block;
240     vertical-align: middle;
241     margin: -2px 0 0 2px;
242     width: 13px;
243     height: 13px;
244 }
245
246 .selector .selector-chosen .help-icon {
247     background: url(../img/icon-unknown-alt.svg) 0 0 no-repeat;
248 }
249
250 .selector .search-label-icon {
251     background: url(../img/search.svg) 0 0 no-repeat;
252     display: inline-block;
253     height: 18px;
254     width: 18px;
255 }
256
257 /* DATE AND TIME */
258
259 p.datetime {
260     line-height: 20px;
261     margin: 0;
262     padding: 0;
263     color: var(--body-quiet-color);
264     font-weight: bold;
265 }

```

```

266
267 .datetime span {
268   white-space: nowrap;
269   font-weight: normal;
270   font-size: 11px;
271   color: var(--body-quiet-color);
272 }
273
274 .datetime input, .form-row .datetime input.vDateField, .form-row .datetime input.vTimeField {
275   margin-left: 5px;
276   margin-bottom: 4px;
277 }
278
279 table p.datetime {
280   font-size: 11px;
281   margin-left: 0;
282   padding-left: 0;
283 }
284
285 .datetimeshortcuts .clock-icon, .datetimeshortcuts .date-icon {
286   position: relative;
287   display: inline-block;
288   vertical-align: middle;
289   height: 16px;
290   width: 16px;
291   overflow: hidden;
292 }
293
294 .datetimeshortcuts .clock-icon {
295   background: url(../img/icon-clock.svg) 0 0 no-repeat;
296 }
297
298 .datetimeshortcuts a:focus .clock-icon,
299 .datetimeshortcuts a:hover .clock-icon {
300   background-position: 0 -16px;
301 }
302
303 .datetimeshortcuts .date-icon {
304   background: url(../img/icon-calendar.svg) 0 0 no-repeat;
305   top: -1px;
306 }
307
308 .datetimeshortcuts a:focus .date-icon,
309 .datetimeshortcuts a:hover .date-icon {
310   background-position: 0 -16px;
311 }
312
313 .timezonewarning {
314   font-size: 11px;
315   color: var(--body-quiet-color);
316 }
317
318 /* URL */
319
320 p.url {
321   line-height: 20px;
322   margin: 0;
323   padding: 0;
324   color: var(--body-quiet-color);
325   font-size: 11px;
326   font-weight: bold;
327 }
328
329 .url a {
330   font-weight: normal;
331 }
332
333 /* FILE UPLOADS */
334
335 p.file-upload {
336   line-height: 20px;
337   margin: 0;
338   padding: 0;
339   color: var(--body-quiet-color);
340   font-size: 11px;
341   font-weight: bold;
342 }
343
344 .aligned p.file-upload {
345   margin-left: 170px;
346 }
347
348 .file-upload a {
349   font-weight: normal;
350 }
351
352 .file-upload .deletelink {
353   margin-left: 5px;
354 }
355
356 span.clearable-file-input label {
357   color: var(--body-fg);
358   font-size: 11px;
359   display: inline;
360   float: none;
361 }
362
363 /* CALENDARS & CLOCKS */
364
365 .calendarbox, .clockbox {
366   margin: 5px auto;
367   font-size: 12px;
368   width: 19em;
369   text-align: center;
370   background: var(--body-bg);
371   color: var(--body-fg);
372   border: 1px solid var(--hairline-color);
373   border-radius: 4px;
374   box-shadow: 0 2px 4px rgba(0, 0, 0, 0.15);
375   overflow: hidden;
376   position: relative;
377 }
378
379 .clockbox {
380   width: auto;

```

```

381 }
382
383 calendar {
384   margin: 0;
385   padding: 0;
386 }
387
388 calendar table {
389   margin: 0;
390   padding: 0;
391   border-collapse: collapse;
392   background: white;
393   width: 100%;
394 }
395
396 calendar caption, .calendarbox h2 {
397   margin: 0;
398   text-align: center;
399   border-top: none;
400   font-weight: 700;
401   font-size: 12px;
402   color: #333;
403   background: var(--accent);
404 }
405
406 calendar th {
407   padding: 8px 5px;
408   background: var(--darkened-bg);
409   border-bottom: 1px solid var(--border-color);
410   font-weight: 400;
411   font-size: 12px;
412   text-align: center;
413   color: var(--body-quiet-color);
414 }
415
416 calendar td {
417   font-weight: 400;
418   font-size: 12px;
419   text-align: center;
420   padding: 0;
421   border-top: 1px solid var(--hairline-color);
422   border-bottom: none;
423 }
424
425 calendar td.selected a {
426   background: var(--primary);
427   color: var(--button-fg);
428 }
429
430 calendar td.noday {
431   background: var(--darkened-bg);
432 }
433
434 calendar td.today a {
435   font-weight: 700;
436 }
437
438 calendar td a, .timelist a {
439   display: block;
440   font-weight: 400;
441   padding: 6px;
442   text-decoration: none;
443   color: var(--body-quiet-color);
444 }
445
446 calendar td a:focus, .timelist a:focus
447 calendar td a:hover, .timelist a:hover {
448   background: var(--primary);
449   color: white;
450 }
451
452 calendar td a:active, .timelist a:active {
453   background: var(--header-bg);
454   color: white;
455 }
456
457 calendarnav {
458   font-size: 10px;
459   text-align: center;
460   color: #ccc;
461   margin: 0;
462   padding: 1px 3px;
463 }
464
465 calendarnav a:link, #calendarnav a:visited
466 #calendarnav a:focus, #calendarnav a:hover {
467   color: var(--body-quiet-color);
468 }
469
470 calendar-shortcuts {
471   background: var(--body-bg);
472   color: var(--body-quiet-color);
473   font-size: 11px;
474   line-height: 11px;
475   border-top: 1px solid var(--hairline-color);
476   padding: 8px 0;
477 }
478
479 calendarbox .calendarnav-previous, .calendarbox .calendarnav-next {
480   display: block;
481   position: absolute;
482   top: 8px;
483   width: 15px;
484   height: 15px;
485   text-indent: -9999px;
486   padding: 0;
487 }
488
489 calendarnav-previous {
490   left: 10px;
491   background: url(../img/calendar-icons.svg) 0 0 no-repeat;
492 }
493
494 calendarbox .calendarnav-previous:focus
495 calendarbox .calendarnav-previous:hover {

```

```

496 background-position: 0 -15px;
497 }
498
499 .calendarnav-next {
500 right: 10px;
501 background: url(../img/calendar-icons.svg) 0 -30px no-repeat;
502 }
503
504 .calendarbox .calendarnav-next:focus,
505 .calendarbox .calendarnav-next:hover {
506 background-position: 0 -45px;
507 }
508
509 .calendar-cancel {
510 margin: 0;
511 padding: 4px 0;
512 font-size: 12px;
513 background: #eee;
514 border-top: 1px solid var(--border-color);
515 color: var(--body-fg);
516 }
517
518 .calendar-cancel:focus, .calendar-cancel:hover {
519 background: #ddd;
520 }
521
522 .calendar-cancel a {
523 color: black;
524 display: block;
525 }
526
527 ul.timelist, .timelist li {
528 list-style-type: none;
529 margin: 0;
530 padding: 0;
531 }
532
533 .timelist a {
534 padding: 2px;
535 }
536
537 /* EDIT INLINE */
538
539 .inline-deletelink {
540 float: right;
541 text-indent: -9999px;
542 background: url(../img/inline-delete.svg) 0 0 no-repeat;
543 width: 16px;
544 height: 16px;
545 border: 0px none;
546 }
547
548 .inline-deletelink:focus, .inline-deletelink:hover {
549 cursor: pointer;
550 }
551
552 /* RELATED WIDGET WRAPPER */
553 .related-widget-wrapper {
554 float: left; /* display properly in form rows with multiple fields */
555 overflow: hidden; /* clear floated contents */
556 }
557
558 .related-widget-wrapper-link {
559 opacity: 0.3;
560 }
561
562 .related-widget-wrapper-link:link {
563 opacity: .8;
564 }
565
566 .related-widget-wrapper-link:link:focus,
567 .related-widget-wrapper-link:link:hover {
568 opacity: 1;
569 }
570
571 select + .related-widget-wrapper-link,
572 .related-widget-wrapper-link + .related-widget-wrapper-link {
573 margin-left: 7px;
574 }

```

**File:** `./static/admin/css/responsive_rtl.css`

```

1  /* TABLETS */
2
3  @media (max-width: 1024px) {
4    [dir="rtl"] .colMS {
5      margin-right: 0;
6    }
7
8    [dir="rtl"] #user-tools {
9      text-align: right;
10   }
11
12   [dir="rtl"] #changelist .actions label {
13     padding-left: 10px;
14     padding-right: 0;
15   }
16
17   [dir="rtl"] #changelist .actions select {
18     margin-left: 0;
19     margin-right: 15px;
20   }
21
22   [dir="rtl"] .change-list .filtered .results,
23   [dir="rtl"] .change-list .filtered .paginator,
24   [dir="rtl"] .filtered #toolbar,
25   [dir="rtl"] .filtered div.xfull,
26   [dir="rtl"] .filtered .actions,
27   [dir="rtl"] #changelist-filter {
28     margin-left: 0;
29   }
30
31   [dir="rtl"] .inline-group ul.tools a.add,
32   [dir="rtl"] .inline-group div.add-row a,
33   [dir="rtl"] .inline-group .tabular tr.add-row td a {
34     padding: 8px 26px 8px 10px;
35     background-position: calc(100% - 8px) 9px;
36   }
37
38   [dir="rtl"] .related-widget-wrapper-link + .selector {
39     margin-right: 0;
40     margin-left: 15px;
41   }
42
43   [dir="rtl"] .selector .selector-filter label {
44     margin-right: 0;
45     margin-left: 8px;
46   }
47
48   [dir="rtl"] .object-tools li {
49     float: right;
50   }
51
52   [dir="rtl"] .object-tools li + li {
53     margin-left: 0;
54     margin-right: 15px;
55   }
56
57   [dir="rtl"] .dashboard .module table td a {
58     padding-left: 0;
59     padding-right: 16px;
60   }
61 }
62
63 /* MOBILE */
64
65 @media (max-width: 767px) {
66   [dir="rtl"] .aligned .related-lookup,
67   [dir="rtl"] .aligned .datetimeshortcuts {
68     margin-left: 0;
69     margin-right: 15px;
70   }
71
72   [dir="rtl"] .aligned ul {
73     margin-right: 0;
74   }
75
76   [dir="rtl"] #changelist-filter {
77     margin-left: 0;
78     margin-right: 0;
79   }
80 }

```

**File: ./static/admin/css/base.css**

```

1  /*
2   DJANGO Admin styles
3  */
4
5  @import url(fonts.css);
6
7  /* VARIABLE DEFINITIONS */
8  :root {
9    --primary: #79aec8;
10   --secondary: #417690;
11   --accent: #f5dd5d;
12   --primary-fg: #fff;
13
14   --body-fg: #333;
15   --body-bg: #fff;
16   --body-quiet-color: #666;
17   --body-loud-color: #000;
18
19   --header-color: #ffc;
20   --header-branding-color: var(--accent);
21   --header-bg: var(--secondary);
22   --header-link-color: var(--primary-fg);
23
24   --breadcrumbs-fg: #c4c4c4;
25   --breadcrumbs-link-fg: var(--body-bg);
26   --breadcrumbs-bg: var(--primary);
27
28   --link-fg: #447e9b;
29   --link-hover-color: #036;

```

```

30 --link-selected-fg: #5b80b2;
31
32 --hairline-color: #e8e8e8;
33 --border-color: #ccc;
34
35 --error-fg: #ba2121;
36
37 --message-success-bg: #dfd;
38 --message-warning-bg: #ffc;
39 --message-error-bg: #ffeef;
40
41 --darkened-bg: #f8f8f8; /* A bit darker than --body-bg */
42 --selected-bg: #e4e4e4; /* E.g. selected table cells */
43 --selected-row: #ffc;
44
45 --button-fg: #fff;
46 --button-bg: var(--primary);
47 --button-hover-bg: #609ab6;
48 --default-button-bg: var(--secondary);
49 --default-button-hover-bg: #205067;
50 --close-button-bg: #888; /* Previously #bbb, contrast 1.92 */
51 --close-button-hover-bg: #747474;
52 --delete-button-bg: #ba2121;
53 --delete-button-hover-bg: #a41515;
54
55 --object-tools-fg: var(--button-fg);
56 --object-tools-bg: var(--close-button-bg);
57 --object-tools-hover-bg: var(--close-button-hover-bg);
58 }
59
60 @media (prefers-color-scheme: dark) {
61 :root {
62 --primary: #264b5d;
63 --primary-fg: #eee;
64
65 --body-fg: #eeeeee;
66 --body-bg: #121212;
67 --body-quiet-color: #e0e0e0;
68 --body-loud-color: #ffffff;
69
70 --breadcrumbs-link-fg: #e0e0e0;
71 --breadcrumbs-bg: var(--primary);
72
73 --link-fg: #81d4fa;
74 --link-hover-color: #4ac1f7;
75 --link-selected-fg: #6f94c6;
76
77 --hairline-color: #272727;
78 --border-color: #353535;
79
80 --error-fg: #e35f5f;
81 --message-success-bg: #006b1b;
82 --message-warning-bg: #583305;
83 --message-error-bg: #570808;
84
85 --darkened-bg: #212121;
86 --selected-bg: #1b1b1b;
87 --selected-row: #00363a;
88
89 --close-button-bg: #333333;
90 --close-button-hover-bg: #666666;
91 }
92 }
93
94 html, body {
95 height: 100%;
96 }
97
98 body {
99 margin: 0;
100 padding: 0;
101 font-size: 14px;
102 font-family: "Roboto", "Lucida Grande", "DejaVu Sans", "Bitstream Vera Sans", Verdana, Arial, sans-serif;
103 color: var(--body-fg);
104 background: var(--body-bg);
105 }
106
107 /* LINKS */
108
109 a:link, a:visited {
110 color: var(--link-fg);
111 text-decoration: none;
112 transition: color 0.15s, background 0.15s;
113 }
114
115 a:focus, a:hover {
116 color: var(--link-hover-color);
117 }
118
119 a:focus {
120 text-decoration: underline;
121 }
122
123 a img {
124 border: none;
125 }
126
127 a.section:link, a.section:visited {
128 color: var(--header-link-color);
129 text-decoration: none;
130 }
131
132 a.section:focus, a.section:hover {
133 text-decoration: underline;
134 }
135
136 /* GLOBAL DEFAULTS */
137
138 p, ol, ul, dl {
139 margin: .2em 0 .8em 0;
140 }
141
142 p {
143 padding: 0;
144 line-height: 140%;

```

```
145 }
146
147 h1,h2,h3,h4,h5 {
148   font-weight: bold;
149 }
150
151 h1 {
152   margin: 0 0 20px;
153   font-weight: 300;
154   font-size: 20px;
155   color: var(--body-quiet-color);
156 }
157
158 h2 {
159   font-size: 16px;
160   margin: 1em 0 .5em 0;
161 }
162
163 h2.subhead {
164   font-weight: normal;
165   margin-top: 0;
166 }
167
168 h3 {
169   font-size: 14px;
170   margin: .8em 0 .3em 0;
171   color: var(--body-quiet-color);
172   font-weight: bold;
173 }
174
175 h4 {
176   font-size: 12px;
177   margin: 1em 0 .8em 0;
178   padding-bottom: 3px;
179 }
180
181 h5 {
182   font-size: 10px;
183   margin: 1.5em 0 .5em 0;
184   color: var(--body-quiet-color);
185   text-transform: uppercase;
186   letter-spacing: 1px;
187 }
188
189 ul > li {
190   list-style-type: square;
191   padding: 1px 0;
192 }
193
194 li ul {
195   margin-bottom: 0;
196 }
197
198 li, dt, dd {
199   font-size: 13px;
200   line-height: 20px;
201 }
202
203 dt {
204   font-weight: bold;
205   margin-top: 4px;
206 }
207
208 dd {
209   margin-left: 0;
210 }
211
212 form {
213   margin: 0;
214   padding: 0;
215 }
216
217 fieldset {
218   margin: 0;
219   min-width: 0;
220   padding: 0;
221   border: none;
222   border-top: 1px solid var(--hairline-color);
223 }
224
225 blockquote {
226   font-size: 11px;
227   color: #777;
228   margin-left: 2px;
229   padding-left: 10px;
230   border-left: 5px solid #ddd;
231 }
232
233 code, pre {
234   font-family: "Bitstream Vera Sans Mono", Monaco, "Courier New", Courier, monospace;
235   color: var(--body-quiet-color);
236   font-size: 12px;
237   overflow-x: auto;
238 }
239
240 pre.literal-block {
241   margin: 10px;
242   background: var(--darkened-bg);
243   padding: 6px 8px;
244 }
245
246 code strong {
247   color: #930;
248 }
249
250 hr {
251   clear: both;
252   color: var(--hairline-color);
253   background-color: var(--hairline-color);
254   height: 1px;
255   border: none;
256   margin: 0;
257   padding: 0;
258   font-size: 1px;
259   line-height: 1px;
```

```

260 }
261
262 /* TEXT STYLES & MODIFIERS */
263
264 .small {
265     font-size: 11px;
266 }
267
268 .mini {
269     font-size: 10px;
270 }
271
272 .help, p.help, form p.help, div.help, form div.help, div.help li {
273     font-size: 11px;
274     color: var(--body-quiet-color);
275 }
276
277 div.help ul {
278     margin-bottom: 0;
279 }
280
281 .help-tooltip {
282     cursor: help;
283 }
284
285 p img, h1 img, h2 img, h3 img, h4 img, td img {
286     vertical-align: middle;
287 }
288
289 .quiet, a.quiet:link, a.quiet:visited {
290     color: var(--body-quiet-color);
291     font-weight: normal;
292 }
293
294 .clear {
295     clear: both;
296 }
297
298 .nowrap {
299     white-space: nowrap;
300 }
301
302 .hidden {
303     display: none;
304 }
305
306 /* TABLES */
307
308 table {
309     border-collapse: collapse;
310     border-color: var(--border-color);
311 }
312
313 td, th {
314     font-size: 13px;
315     line-height: 16px;
316     border-bottom: 1px solid var(--hairline-color);
317     vertical-align: top;
318     padding: 8px;
319 }
320
321 th {
322     font-weight: 600;
323     text-align: left;
324 }
325
326 thead th,
327 tfoot td {
328     color: var(--body-quiet-color);
329     padding: 5px 10px;
330     font-size: 11px;
331     background: var(--body-bg);
332     border: none;
333     border-top: 1px solid var(--hairline-color);
334     border-bottom: 1px solid var(--hairline-color);
335 }
336
337 tfoot td {
338     border-bottom: none;
339     border-top: 1px solid var(--hairline-color);
340 }
341
342 thead th.required {
343     color: var(--body-loud-color);
344 }
345
346 tr.alt {
347     background: var(--darkened-bg);
348 }
349
350 tr:nth-child(odd), .row-form-errors {
351     background: var(--body-bg);
352 }
353
354 tr:nth-child(even),
355 tr:nth-child(even) .errorlist,
356 tr:nth-child(odd) + .row-form-errors,
357 tr:nth-child(odd) + .row-form-errors .errorlist {
358     background: var(--darkened-bg);
359 }
360
361 /* SORTABLE TABLES */
362
363 thead th {
364     padding: 5px 10px;
365     line-height: normal;
366     text-transform: uppercase;
367     background: var(--darkened-bg);
368 }
369
370 thead th a:link, thead th a:visited {
371     color: var(--body-quiet-color);
372 }
373
374 thead th.sorted {

```



```

375     background: var(--selected-bg);
376 }
377
378 thead th.sorted .text {
379     padding-right: 42px;
380 }
381
382 table thead th .text span {
383     padding: 8px 10px;
384     display: block;
385 }
386
387 table thead th .text a {
388     display: block;
389     cursor: pointer;
390     padding: 8px 10px;
391 }
392
393 table thead th .text a:focus, table thead th .text a:hover {
394     background: var(--selected-bg);
395 }
396
397 thead th.sorted a.sortremove {
398     visibility: hidden;
399 }
400
401 table thead th.sorted:hover a.sortremove {
402     visibility: visible;
403 }
404
405 table thead th.sorted .sortoptions {
406     display: block;
407     padding: 9px 5px 0 5px;
408     float: right;
409     text-align: right;
410 }
411
412 table thead th.sorted .sortpriority {
413     font-size: .8em;
414     min-width: 12px;
415     text-align: center;
416     vertical-align: 3px;
417     margin-left: 2px;
418     margin-right: 2px;
419 }
420
421 table thead th.sorted .sortoptions a {
422     position: relative;
423     width: 14px;
424     height: 14px;
425     display: inline-block;
426     background: url(../img/sorting-icons.svg) 0 0 no-repeat;
427     background-size: 14px auto;
428 }
429
430 table thead th.sorted .sortoptions a.sortremove {
431     background-position: 0 0;
432 }
433
434 table thead th.sorted .sortoptions a.sortremove:after {
435     content: '\';
436     position: absolute;
437     top: -6px;
438     left: 3px;
439     font-weight: 200;
440     font-size: 18px;
441     color: var(--body-quiet-color);
442 }
443
444 table thead th.sorted .sortoptions a.sortremove:focus:after,
445 table thead th.sorted .sortoptions a.sortremove:hover:after {
446     color: var(--link-fg);
447 }
448
449 table thead th.sorted .sortoptions a.sortremove:focus,
450 table thead th.sorted .sortoptions a.sortremove:hover {
451     background-position: 0 -14px;
452 }
453
454 table thead th.sorted .sortoptions a.sortoptions a.sortremove:after {
455     background-position: 0 -28px;
456 }
457
458 table thead th.sorted .sortoptions a.sortoptions a.sortremove:after {
459     background-position: 0 -42px;
460 }
461
462 table thead th.sorted .sortoptions a.sortoptions a.sortremove:after {
463     background-position: 0 -56px;
464 }
465
466 table thead th.sorted .sortoptions a.sortoptions a.sortremove:after {
467     background-position: 0 -70px;
468 }
469
470 /* FORM DEFAULTS */
471
472 input, textarea, select, .form-row p, form .button {
473     margin: 2px 0;
474     padding: 2px 3px;
475     vertical-align: middle;
476     font-family: "Roboto", "Lucida Grande", Verdana, Arial, sans-serif;
477     font-weight: normal;
478     font-size: 13px;
479 }
480
481 .form-row div.help {
482     padding: 2px 3px;
483 }
484
485 textarea {
486     vertical-align: top;
487 }

```

```

490
491 input[type=text], input[type=password], input[type=email], input[type=url],
492 input[type=number], input[type=tel], textarea, select, .vTextField {
493     border: 1px solid var(--border-color);
494     border-radius: 4px;
495     padding: 5px 6px;
496     margin-top: 0;
497     color: var(--body-fg);
498     background-color: var(--body-bg);
499 }
500
501 input[type=text]:focus, input[type=password]:focus, input[type=email]:focus,
502 input[type=url]:focus, input[type=number]:focus, input[type=tel]:focus,
503 textarea:focus, select:focus, .vTextField:focus {
504     border-color: var(--body-quiet-color);
505 }
506
507 select {
508     height: 30px;
509 }
510
511 select[multiple] {
512     /* Allow HTML size attribute to override the height in the rule above. */
513     height: auto;
514     min-height: 150px;
515 }
516
517 /* FORM BUTTONS */
518
519 .button, input[type=submit], input[type=button], .submit-row input, a.button {
520     background: var(--button-bg);
521     padding: 10px 15px;
522     border: none;
523     border-radius: 4px;
524     color: var(--button-fg);
525     cursor: pointer;
526     transition: background 0.15s;
527 }
528
529 a.button {
530     padding: 4px 5px;
531 }
532
533 .button:active, input[type=submit]:active, input[type=button]:active,
534 .button:focus, input[type=submit]:focus, input[type=button]:focus,
535 .button:hover, input[type=submit]:hover, input[type=button]:hover {
536     background: var(--button-hover-bg);
537 }
538
539 .button[disabled], input[type=submit][disabled], input[type=button][disabled] {
540     opacity: 0.4;
541 }
542
543 .button.default, input[type=submit].default, .submit-row input.default {
544     float: right;
545     border: none;
546     font-weight: 400;
547     background: var(--default-button-bg);
548 }
549
550 .button.default:active, input[type=submit].default:active,
551 .button.default:focus, input[type=submit].default:focus,
552 .button.default:hover, input[type=submit].default:hover {
553     background: var(--default-button-hover-bg);
554 }
555
556 .button[disabled].default,
557 input[type=submit][disabled].default,
558 input[type=button][disabled].default {
559     opacity: 0.4;
560 }
561
562
563 /* MODULES */
564
565 .module {
566     border: none;
567     margin-bottom: 30px;
568     background: var(--body-bg);
569 }
570
571 .module p, .module ul, .module h3, .module h4, .module dl, .module pre {
572     padding-left: 10px;
573     padding-right: 10px;
574 }
575
576 .module blockquote {
577     margin-left: 12px;
578 }
579
580 .module ul, .module ol {
581     margin-left: 1.5em;
582 }
583
584 .module h3 {
585     margin-top: .6em;
586 }
587
588 .module h2, .module caption, .inline-group h2 {
589     margin: 0;
590     padding: 8px;
591     font-weight: 400;
592     font-size: 13px;
593     text-align: left;
594     background: var(--primary);
595     color: var(--header-link-color);
596 }
597
598 .module caption,
599 .inline-group h2 {
600     font-size: 12px;
601     letter-spacing: 0.5px;
602     text-transform: uppercase;
603 }
604

```

```

605 .module table {
606   border-collapse: collapse;
607 }
608
609 /* MESSAGES & ERRORS */
610
611 ul.messageList {
612   padding: 0;
613   margin: 0;
614 }
615
616 ul.messageList li {
617   display: block;
618   font-weight: 400;
619   font-size: 13px;
620   padding: 10px 10px 10px 65px;
621   margin: 0 0 10px 0;
622   background: var(--message-success-bg) url(..img/icon-yes.svg) 40px 12px no-repeat;
623   background-size: 16px auto;
624   color: var(--body-fg);
625 }
626
627 ul.messageList li.warning {
628   background: var(--message-warning-bg) url(..img/icon-alert.svg) 40px 14px no-repeat;
629   background-size: 14px auto;
630 }
631
632 ul.messageList li.error {
633   background: var(--message-error-bg) url(..img/icon-no.svg) 40px 12px no-repeat;
634   background-size: 16px auto;
635 }
636
637 .errornote {
638   font-size: 14px;
639   font-weight: 700;
640   display: block;
641   padding: 10px 12px;
642   margin: 0 0 10px 0;
643   color: var(--error-fg);
644   border: 1px solid var(--error-fg);
645   border-radius: 4px;
646   background-color: var(--body-bg);
647   background-position: 5px 12px;
648   overflow-wrap: break-word;
649 }
650
651 ul.errorlist {
652   margin: 0 0 4px;
653   padding: 0;
654   color: var(--error-fg);
655   background: var(--body-bg);
656 }
657
658 ul.errorlist li {
659   font-size: 13px;
660   display: block;
661   margin-bottom: 4px;
662   overflow-wrap: break-word;
663 }
664
665 ul.errorlist li:first-child {
666   margin-top: 0;
667 }
668
669 ul.errorlist li a {
670   color: inherit;
671   text-decoration: underline;
672 }
673
674 td ul.errorlist {
675   margin: 0;
676   padding: 0;
677 }
678
679 td ul.errorlist li {
680   margin: 0;
681 }
682
683 .form-row.errors {
684   margin: 0;
685   border: none;
686   border-bottom: 1px solid var(--hairline-color);
687   background: none;
688 }
689
690 .form-row.errors ul.errorlist li {
691   padding-left: 0;
692 }
693
694 .errors input, .errors select, .errors textarea,
695 td ul.errorlist + input, td ul.errorlist + select, td ul.errorlist + textarea {
696   border: 1px solid var(--error-fg);
697 }
698
699 .description {
700   font-size: 12px;
701   padding: 5px 0 0 12px;
702 }
703
704 /* BREADCRUMBS */
705
706 div.breadcrumbs {
707   background: var(--breadcrumbs-bg);
708   padding: 10px 40px;
709   border: none;
710   color: var(--breadcrumbs-fg);
711   text-align: left;
712 }
713
714 div.breadcrumbs a {
715   color: var(--breadcrumbs-link-fg);
716 }
717
718 div.breadcrumbs a:focus, div.breadcrumbs a:hover {
719   color: var(--breadcrumbs-fg);

```

```

720 }
721
722 /* ACTION ICONS */
723
724 .viewlink, .inlineviewlink {
725     padding-left: 16px;
726     background: url(..img/icon-viewlink.svg) 0 1px no-repeat;
727 }
728
729 .addlink {
730     padding-left: 16px;
731     background: url(..img/icon-addlink.svg) 0 1px no-repeat;
732 }
733
734 .changelink, .inlinechangelink {
735     padding-left: 16px;
736     background: url(..img/icon-changelink.svg) 0 1px no-repeat;
737 }
738
739 .deletelink {
740     padding-left: 16px;
741     background: url(..img/icon-deletelink.svg) 0 1px no-repeat;
742 }
743
744 a.deletelink:link, a.deletelink:visited {
745     color: #CC3434; /* XXX Probably unused? */
746 }
747
748 a.deletelink:focus, a.deletelink:hover {
749     color: #993333; /* XXX Probably unused? */
750     text-decoration: none;
751 }
752
753 /* OBJECT TOOLS */
754
755 .object-tools {
756     font-size: 10px;
757     font-weight: bold;
758     padding-left: 0;
759     float: right;
760     position: relative;
761     margin-top: -48px;
762 }
763
764 .object-tools li {
765     display: block;
766     float: left;
767     margin-left: 5px;
768     height: 16px;
769 }
770
771 .object-tools a {
772     border-radius: 15px;
773 }
774
775 .object-tools a:link, .object-tools a:visited {
776     display: block;
777     float: left;
778     padding: 3px 12px;
779     background: var(--object-tools-bg);
780     color: var(--object-tools-fg);
781     font-weight: 400;
782     font-size: 11px;
783     text-transform: uppercase;
784     letter-spacing: 0.5px;
785 }
786
787 .object-tools a:focus, .object-tools a:hover {
788     background-color: var(--object-tools-hover-bg);
789 }
790
791 .object-tools a:focus {
792     text-decoration: none;
793 }
794
795 .object-tools a.viewsitelink, .object-tools a.addlink {
796     background-repeat: no-repeat;
797     background-position: right 7px center;
798     padding-right: 26px;
799 }
800
801 .object-tools a.viewsitelink {
802     background-image: url(..img/tooltag-arrowright.svg);
803 }
804
805 .object-tools a.addlink {
806     background-image: url(..img/tooltag-add.svg);
807 }
808
809 /* OBJECT HISTORY */
810
811 table#change-history {
812     width: 100%;
813 }
814
815 table#change-history tbody th {
816     width: 16em;
817 }
818
819 /* PAGE STRUCTURE */
820
821 #container {
822     position: relative;
823     width: 100%;
824     min-width: 980px;
825     padding: 0;
826     display: flex;
827     flex-direction: column;
828     height: 100%;
829 }
830
831 #container > div {
832     flex-shrink: 0;
833 }
834

```

```

835 #container > .main {
836   display: flex;
837   flex: 1 0 auto;
838 }
839
840 .main > .content {
841   flex: 1 0;
842   max-width: 100%;
843 }
844
845 #content {
846   padding: 20px 40px;
847 }
848
849 .dashboard #content {
850   width: 600px;
851 }
852
853 #content-main {
854   float: left;
855   width: 100%;
856 }
857
858 #content-related {
859   float: right;
860   width: 260px;
861   position: relative;
862   margin-right: -300px;
863 }
864
865 #footer {
866   clear: both;
867   padding: 10px;
868 }
869
870 /* COLUMN TYPES */
871
872 .colMS {
873   margin-right: 300px;
874 }
875
876 .colSM {
877   margin-left: 300px;
878 }
879
880 .colSM #content-related {
881   float: left;
882   margin-right: 0;
883   margin-left: -300px;
884 }
885
886 .colSM #content-main {
887   float: right;
888 }
889
890 .popup .colM {
891   width: auto;
892 }
893
894 /* HEADER */
895
896 #header {
897   width: auto;
898   height: auto;
899   display: flex;
900   justify-content: space-between;
901   align-items: center;
902   padding: 10px 40px;
903   background: var(--header-bg);
904   color: var(--header-color);
905   overflow: hidden;
906 }
907
908 #header a:link, #header a:visited {
909   color: var(--header-link-color);
910 }
911
912 #header a:focus, #header a:hover {
913   text-decoration: underline;
914 }
915
916 #branding {
917   float: left;
918 }
919
920 #branding h1 {
921   padding: 0;
922   margin: 0 20px 0 0;
923   font-weight: 300;
924   font-size: 24px;
925   color: var(--accent);
926 }
927
928 #branding h1, #branding h1 a:link, #branding h1 a:visited {
929   color: var(--accent);
930 }
931
932 #branding h2 {
933   padding: 0 10px;
934   font-size: 14px;
935   margin: -8px 0 8px 0;
936   font-weight: normal;
937   color: var(--header-color);
938 }
939
940 #branding a:hover {
941   text-decoration: none;
942 }
943
944 #user-tools {
945   float: right;
946   padding: 0;
947   margin: 0 0 20px 0;
948   font-weight: 300;
949   font-size: 11px;

```

```

950 letter-spacing: 0.5px;
951 text-transform: uppercase;
952 text-align: right;
953 }
954
955 #user-tools a {
956 border-bottom: 1px solid rgba(255, 255, 255, 0.25);
957 }
958
959 #user-tools a:focus, #user-tools a:hover {
960 text-decoration: none;
961 border-bottom-color: var(--primary);
962 color: var(--primary);
963 }
964
965 /* SIDEBAR */
966
967 #content-related {
968 background: var(--darkened-bg);
969 }
970
971 #content-related .module {
972 background: none;
973 }
974
975 #content-related h3 {
976 color: var(--body-quiet-color);
977 padding: 0 16px;
978 margin: 0 0 16px;
979 }
980
981 #content-related h4 {
982 font-size: 13px;
983 }
984
985 #content-related p {
986 padding-left: 16px;
987 padding-right: 16px;
988 }
989
990 #content-related .actionlist {
991 padding: 0;
992 margin: 16px;
993 }
994
995 #content-related .actionlist li {
996 line-height: 1.2;
997 margin-bottom: 10px;
998 padding-left: 18px;
999 }
1000
1001 #content-related .module h2 {
1002 background: none;
1003 padding: 16px;
1004 margin-bottom: 16px;
1005 border-bottom: 1px solid var(--hairline-color);
1006 font-size: 18px;
1007 color: var(--body-fg);
1008 }
1009
1010 .delete-confirmation form input[type="submit"] {
1011 background: var(--delete-button-bg);
1012 border-radius: 4px;
1013 padding: 10px 15px;
1014 color: var(--button-fg);
1015 }
1016
1017 .delete-confirmation form input[type="submit"]:active,
1018 .delete-confirmation form input[type="submit"]:focus,
1019 .delete-confirmation form input[type="submit"]:hover {
1020 background: var(--delete-button-hover-bg);
1021 }
1022
1023 .delete-confirmation form .cancel-link {
1024 display: inline-block;
1025 vertical-align: middle;
1026 height: 15px;
1027 line-height: 15px;
1028 border-radius: 4px;
1029 padding: 10px 15px;
1030 color: var(--button-fg);
1031 background: var(--close-button-bg);
1032 margin: 0 0 0 10px;
1033 }
1034
1035 .delete-confirmation form .cancel-link:active,
1036 .delete-confirmation form .cancel-link:focus,
1037 .delete-confirmation form .cancel-link:hover {
1038 background: var(--close-button-hover-bg);
1039 }
1040
1041 /* POPUP */
1042 .popup #content {
1043 padding: 20px;
1044 }
1045
1046 .popup #container {
1047 min-width: 0;
1048 }
1049
1050 .popup #header {
1051 padding: 10px 20px;
1052 }

```

**File: ./static/admin/css/forms.css**

```
1  @import url('widgets.css');
2
3  /* FORM ROWS */
4
5  .form-row {
6      overflow: hidden;
7      padding: 10px;
8      font-size: 13px;
9      border-bottom: 1px solid var(--hairline-color);
10 }
11
12 .form-row img, .form-row input {
13     vertical-align: middle;
14 }
15
16 .form-row label input[type="checkbox"] {
17     margin-top: 0;
18     vertical-align: 0;
19 }
20
21 form .form-row p {
22     padding-left: 0;
23 }
24
25 /* FORM LABELS */
26
27 label {
28     font-weight: normal;
29     color: var(--body-quiet-color);
30     font-size: 13px;
31 }
32
33 .required label, label.required {
34     font-weight: bold;
35     color: var(--body-fg);
36 }
37
38 /* RADIO BUTTONS */
39
40 form ul.radiolist li {
41     list-style-type: none;
42 }
43
44 form ul.radiolist label {
45     float: none;
46     display: inline;
47 }
48
49 form ul.radiolist input[type="radio"] {
50     margin: -2px 4px 0 0;
51     padding: 0;
52 }
53
54 form ul.inline {
55     margin-left: 0;
56     padding: 0;
57 }
58
59 form ul.inline li {
60     float: left;
61     padding-right: 7px;
62 }
63
64 /* ALIGNED FIELDSETS */
65
66 .aligned label {
67     display: block;
68     padding: 4px 10px 0 0;
69     float: left;
70     width: 160px;
71     word-wrap: break-word;
72     line-height: 1;
73 }
74
75 .aligned label:not(.vCheckboxLabel):after {
76     content: "";
77     display: inline-block;
78     vertical-align: middle;
79     height: 26px;
80 }
81
82 .aligned label + p, .aligned label + div.help, .aligned label + div.readonly {
83     padding: 6px 0;
84     margin-top: 0;
85     margin-bottom: 0;
86     margin-left: 170px;
87     overflow-wrap: break-word;
88 }
89
90 .aligned ul label {
91     display: inline;
92     float: none;
93     width: auto;
94 }
95
96 .aligned .form-row input {
97     margin-bottom: 0;
98 }
99
100 .colMS .aligned .vLargeTextField, .colMS .aligned .vXMLLargeTextField {
101     width: 350px;
102 }
103
104 form .aligned ul {
105     margin-left: 160px;
106     padding-left: 10px;
107 }
108
109 form .aligned ul.radiolist {
```

```

110 display: inline-block;
111 margin: 0;
112 padding: 0;
113 }
114
115 form .aligned p.help,
116 form .aligned div.help {
117     clear: left;
118     margin-top: 0;
119     margin-left: 160px;
120     padding-left: 10px;
121 }
122
123 form .aligned label + p.help,
124 form .aligned label + div.help {
125     margin-left: 0;
126     padding-left: 0;
127 }
128
129 form .aligned p.help:last-child,
130 form .aligned div.help:last-child {
131     margin-bottom: 0;
132     padding-bottom: 0;
133 }
134
135 form .aligned input + p.help,
136 form .aligned textarea + p.help,
137 form .aligned select + p.help,
138 form .aligned input + div.help,
139 form .aligned textarea + div.help,
140 form .aligned select + div.help {
141     margin-left: 160px;
142     padding-left: 10px;
143 }
144
145 form .aligned ul li {
146     list-style: none;
147 }
148
149 form .aligned table p {
150     margin-left: 0;
151     padding-left: 0;
152 }
153
154 .aligned .vCheckboxLabel {
155     float: none;
156     width: auto;
157     display: inline-block;
158     vertical-align: -3px;
159     padding: 0 0 5px 5px;
160 }
161
162 .aligned .vCheckboxLabel + p.help,
163 .aligned .vCheckboxLabel + div.help {
164     margin-top: -4px;
165 }
166
167 colM .aligned .vLargeTextField, .colM .aligned .vXMLLargeTextField {
168     width: 610px;
169 }
170
171 .checkbox-row p.help,
172 .checkbox-row div.help {
173     margin-left: 0;
174     padding-left: 0;
175 }
176
177 fieldset .fieldBox {
178     float: left;
179     margin-right: 20px;
180 }
181
182 /* WIDE FIELDSETS */
183
184 .wide label {
185     width: 200px;
186 }
187
188 form .wide p,
189 form .wide input + p.help,
190 form .wide input + div.help {
191     margin-left: 200px;
192 }
193
194 form .wide p.help,
195 form .wide div.help {
196     padding-left: 38px;
197 }
198
199 form div.help ul {
200     padding-left: 0;
201     margin-left: 0;
202 }
203
204 colM fieldset.wide .vLargeTextField, .colM fieldset.wide .vXMLLargeTextField {
205     width: 450px;
206 }
207
208 /* COLLAPSED FIELDSETS */
209
210 fieldset.collapsed * {
211     display: none;
212 }
213
214 fieldset.collapsed h2, fieldset.collapsed {
215     display: block;
216 }
217
218 fieldset.collapsed {
219     border: 1px solid var(--hairline-color);
220     border-radius: 4px;
221     overflow: hidden;
222 }
223
224 fieldset.collapsed h2 {

```



```

225 background: var(--darkened-bg);
226 color: var(--body-quiet-color);
227 }
228
229 fieldset .collapse-toggle {
230   color: var(--header-link-color);
231 }
232
233 fieldset.collapsed .collapse-toggle {
234   background: transparent;
235   display: inline;
236   color: var(--link-fg);
237 }
238
239 /* MONOSPACE TEXTAREAS */
240
241 fieldset.monospace textarea {
242   font-family: "Bitstream Vera Sans Mono", Monaco, "Courier New", Courier, monospace;
243 }
244
245 /* SUBMIT ROW */
246
247 .submit-row {
248   padding: 12px 14px;
249   margin: 0 0 20px;
250   background: var(--darkened-bg);
251   border: 1px solid var(--hairline-color);
252   border-radius: 4px;
253   text-align: right;
254   overflow: hidden;
255 }
256
257 body.popup .submit-row {
258   overflow: auto;
259 }
260
261 .submit-row input {
262   height: 35px;
263   line-height: 15px;
264   margin: 0 0 0 5px;
265 }
266
267 .submit-row input.default {
268   margin: 0 0 0 8px;
269   text-transform: uppercase;
270 }
271
272 .submit-row p {
273   margin: 0.3em;
274 }
275
276 .submit-row p.deletelink-box {
277   float: left;
278   margin: 0;
279 }
280
281 .submit-row a.deletelink {
282   display: block;
283   background: var(--delete-button-bg);
284   border-radius: 4px;
285   padding: 10px 15px;
286   height: 15px;
287   line-height: 15px;
288   color: var(--button-fg);
289 }
290
291 .submit-row a.closetlink {
292   display: inline-block;
293   background: var(--close-button-bg);
294   border-radius: 4px;
295   padding: 10px 15px;
296   height: 15px;
297   line-height: 15px;
298   margin: 0 0 0 5px;
299   color: var(--button-fg);
300 }
301
302 .submit-row a.deletelink:focus,
303 .submit-row a.deletelink:hover,
304 .submit-row a.deletelink:active {
305   background: var(--delete-button-hover-bg);
306 }
307
308 .submit-row a.closetlink:focus,
309 .submit-row a.closetlink:hover,
310 .submit-row a.closetlink:active {
311   background: var(--close-button-hover-bg);
312 }
313
314 /* CUSTOM FORM FIELDS */
315
316 vSelectMultipleField {
317   vertical-align: top;
318 }
319
320 vCheckboxField {
321   border: none;
322 }
323
324 vDateField, vTimeField {
325   margin-right: 2px;
326   margin-bottom: 4px;
327 }
328
329 vDateField {
330   min-width: 6.85em;
331 }
332
333 vTimeField {
334   min-width: 4.7em;
335 }
336
337 vURLField {
338   width: 30em;
339 }

```

```

340
341 .vLargeTextField, .vXMLLargeTextField {
342     width: 48em;
343 }
344
345 flatpages-flatpage #id_content {
346     height: 40.2em;
347 }
348
349 module table .vPositiveSmallIntegerField {
350     width: 2.2em;
351 }
352
353 vTextField, .vUUIDField {
354     width: 20em;
355 }
356
357 vIntegerField {
358     width: 5em;
359 }
360
361 vBigIntegerField {
362     width: 10em;
363 }
364
365 vForeignKeyRawIdAdminField {
366     width: 5em;
367 }
368
369 /* INLINES */
370
371 .inline-group {
372     padding: 0;
373     margin: 0 0 30px;
374 }
375
376 .inline-group thead th {
377     padding: 8px 10px;
378 }
379
380 .inline-group .aligned label {
381     width: 160px;
382 }
383
384 .inline-related {
385     position: relative;
386 }
387
388 .inline-related h3 {
389     margin: 0;
390     color: var(--body-quiet-color);
391     padding: 5px;
392     font-size: 13px;
393     background: var(--darkened-bg);
394     border-top: 1px solid var(--hairline-color);
395     border-bottom: 1px solid var(--hairline-color);
396 }
397
398 .inline-related h3 span.delete {
399     float: right;
400 }
401
402 .inline-related h3 span.delete label {
403     margin-left: 2px;
404     font-size: 11px;
405 }
406
407 .inline-related fieldset {
408     margin: 0;
409     background: var(--body-bg);
410     border: none;
411     width: 100%;
412 }
413
414 .inline-related fieldset.module h3 {
415     margin: 0;
416     padding: 2px 5px 3px 5px;
417     font-size: 11px;
418     text-align: left;
419     font-weight: bold;
420     background: #bcd;
421     color: var(--body-bg);
422 }
423
424 .inline-group .tabular fieldset.module {
425     border: none;
426 }
427
428 .inline-related .tabular fieldset.module table {
429     width: 100%;
430     overflow-x: scroll;
431 }
432
433 .last-related fieldset {
434     border: none;
435 }
436
437 .inline-group .tabular tr.has_original td {
438     padding-top: 2em;
439 }
440
441 .inline-group .tabular tr td.original {
442     padding: 2px 0 0 0;
443     width: 0;
444     _position: relative;
445 }
446
447 .inline-group .tabular th.original {
448     width: 0px;
449     padding: 0;
450 }
451
452 .inline-group .tabular td.original p {
453     position: absolute;
454     left: 0;

```

```

455 height: 1.1em;
456 padding: 2px 9px;
457 overflow: hidden;
458 font-size: 9px;
459 font-weight: bold;
460 color: var(--body-quiet-color);
461 _width: 700px;
462 }
463
464 .inline-group ul.tools {
465 padding: 0;
466 margin: 0;
467 list-style: none;
468 }
469
470 .inline-group ul.tools li {
471 display: inline;
472 padding: 0 5px;
473 }
474
475 .inline-group div.add-row,
476 .inline-group .tabular tr.add-row td {
477 color: var(--body-quiet-color);
478 background: var(--darkened-bg);
479 padding: 8px 10px;
480 border-bottom: 1px solid var(--hairline-color);
481 }
482
483 .inline-group .tabular tr.add-row td {
484 padding: 8px 10px;
485 border-bottom: 1px solid var(--hairline-color);
486 }
487
488 .inline-group ul.tools a.add,
489 .inline-group div.add-row a,
490 .inline-group .tabular tr.add-row td a {
491 background: url(../img/icon-addlink.svg) 0 1px no-repeat;
492 padding-left: 16px;
493 font-size: 12px;
494 }
495
496 .empty-form {
497 display: none;
498 }
499
500 /* RELATED FIELD ADD ONE / LOOKUP */
501
502 .related-lookup {
503 margin-left: 5px;
504 display: inline-block;
505 vertical-align: middle;
506 background-repeat: no-repeat;
507 background-size: 14px;
508 }
509
510 .related-lookup {
511 width: 16px;
512 height: 16px;
513 background-image: url(../img/search.svg);
514 }
515
516 .form .related-widget-wrapper ul {
517 display: inline-block;
518 margin-left: 0;
519 padding-left: 0;
520 }
521
522 .clearable-file-input input {
523 margin-top: 0;
524 }

```

**File:** `./static/admin/css/login.css`

```

1  /* LOGIN FORM */
2
3  .login {
4    background: var(--darkened-bg);
5    height: auto;
6  }
7
8  .login #header {
9    height: auto;
10   padding: 15px 16px;
11   justify-content: center;
12 }
13
14 .login #header h1 {
15   font-size: 18px;
16   margin: 0;
17 }
18
19 .login #header h1 a {
20   color: var(--header-link-color);
21 }
22
23 .login #content {
24   padding: 20px 20px 0;
25 }
26
27 .login #container {
28   background: var(--body-bg);
29   border: 1px solid var(--hairline-color);
30   border-radius: 4px;
31   overflow: hidden;
32   width: 28em;
33   min-width: 300px;
34   margin: 100px auto;
35   height: auto;
36 }
37
38 .login .form-row {
39   padding: 4px 0;
40 }
41
42 .login .form-row label {
43   display: block;
44   line-height: 2em;
45 }
46
47 .login .form-row #id_username, .login .form-row #id_password {
48   padding: 8px;
49   width: 100%;
50   box-sizing: border-box;
51 }
52
53 .login .submit-row {
54   padding: 1em 0 0 0;
55   margin: 0;
56   text-align: center;
57 }
58
59 .login .password-reset-link {
60   text-align: center;
61 }

```

**File: ./static/admin/css/nav\_sidebar.css**

```

1  .sticky {
2    position: sticky;
3    top: 0;
4    max-height: 100vh;
5  }
6
7  .toggle-nav-sidebar {
8    z-index: 20;
9    left: 0;
10   display: flex;
11   align-items: center;
12   justify-content: center;
13   flex: 0 0 23px;
14   width: 23px;
15   border: 0;
16   border-right: 1px solid var(--hairline-color);
17   background-color: var(--body-bg);
18   cursor: pointer;
19   font-size: 20px;
20   color: var(--link-fg);
21   padding: 0;
22 }
23
24 [dir="rtl"] .toggle-nav-sidebar {
25   border-left: 1px solid var(--hairline-color);
26   border-right: 0;
27 }
28
29 .toggle-nav-sidebar:hover,
30 .toggle-nav-sidebar:focus {
31   background-color: var(--darkened-bg);
32 }
33
34 #nav-sidebar {
35   z-index: 15;
36   flex: 0 0 275px;
37   left: -276px;
38   margin-left: -276px;
39   border-top: 1px solid transparent;
40   border-right: 1px solid var(--hairline-color);
41   background-color: var(--body-bg);
42   overflow: auto;
43 }
44
45 [dir="rtl"] #nav-sidebar {
46   border-left: 1px solid var(--hairline-color);
47   border-right: 0;
48   left: 0;

```

```

49     margin-left: 0;
50     right: -276px;
51     margin-right: -276px;
52 }
53
54 .toggle-nav-sidebar::before {
55     content: '\00BB';
56 }
57
58 .main.shifted .toggle-nav-sidebar::before {
59     content: '\00AB';
60 }
61
62 .main.shifted > #nav-sidebar {
63     margin-left: 0;
64 }
65
66 [dir="rtl"] .main.shifted > #nav-sidebar {
67     margin-right: 0;
68 }
69
70 #nav-sidebar .module th {
71     width: 100%;
72     overflow-wrap: anywhere;
73 }
74
75 #nav-sidebar .module th,
76 #nav-sidebar .module caption {
77     padding-left: 16px;
78 }
79
80 #nav-sidebar .module td {
81     white-space: nowrap;
82 }
83
84 [dir="rtl"] #nav-sidebar .module th,
85 [dir="rtl"] #nav-sidebar .module caption {
86     padding-left: 8px;
87     padding-right: 16px;
88 }
89
90 #nav-sidebar .current-app .section:link
91 #nav-sidebar .current-app .section:visited {
92     color: var(--header-color);
93     font-weight: bold;
94 }
95
96 #nav-sidebar .current-model {
97     background: var(--selected-row);
98 }
99
100 .main > #nav-sidebar + .content {
101     max-width: calc(100% - 23px);
102 }
103
104 .main.shifted > #nav-sidebar + .content {
105     max-width: calc(100% - 299px);
106 }
107
108 @media (max-width: 767px) {
109     #nav-sidebar, #toggle-nav-sidebar {
110         display: none;
111     }
112
113     .main > #nav-sidebar + .content,
114     .main.shifted > #nav-sidebar + .content {
115         max-width: 100%;
116     }
117 }
118
119 #nav-filter {
120     width: 100%;
121     box-sizing: border-box;
122     padding: 2px 5px;
123     margin: 5px 0;
124     border: 1px solid var(--border-color);
125     background-color: var(--darkened-bg);
126     color: var(--body-fg);
127 }
128
129 #nav-filter:focus {
130     border-color: var(--body-quiet-color);
131 }
132
133 #nav-filter.no-results {
134     background: var(--message-error-bg);
135 }
136
137 #nav-sidebar table {
138     width: 100%;
139 }

```

**File:** `./static/admin/css/dashboard.css`

```

1  /* DASHBOARD */
2
3  .dashboard .module table th {
4    width: 100%;
5  }
6
7  .dashboard .module table td {
8    white-space: nowrap;
9  }
10
11 .dashboard .module table td a {
12   display: block;
13   padding-right: .6em;
14 }
15
16 /* RECENT ACTIONS MODULE */
17
18 .module ul.actionlist {
19   margin-left: 0;
20 }
21
22 ul.actionlist li {
23   list-style-type: none;
24   overflow: hidden;
25   text-overflow: ellipsis;
26 }

```

**File: ./static/admin/css/autocomplete.css**

```

1  select.admin-autocomplete {
2    width: 20em;
3  }
4
5  .select2-container--admin-autocomplete.select2-container {
6    min-height: 30px;
7  }
8
9  .select2-container--admin-autocomplete .select2-selection--single,
10 .select2-container--admin-autocomplete .select2-selection--multiple {
11   min-height: 30px;
12   padding: 0;
13 }
14
15 .select2-container--admin-autocomplete.select2-container--focus .select2-selection,
16 .select2-container--admin-autocomplete.select2-container--open .select2-selection {
17   border-color: var(--body-quiet-color);
18   min-height: 30px;
19 }
20
21 .select2-container--admin-autocomplete.select2-container--focus .select2-selection.select2-selection--single,
22 .select2-container--admin-autocomplete.select2-container--open .select2-selection.select2-selection--single {
23   padding: 0;
24 }
25
26 .select2-container--admin-autocomplete.select2-container--focus .select2-selection.select2-selection--multiple,
27 .select2-container--admin-autocomplete.select2-container--open .select2-selection.select2-selection--multiple {
28   padding: 0;
29 }
30
31 .select2-container--admin-autocomplete .select2-selection--single {
32   background-color: var(--body-bg);
33   border: 1px solid var(--border-color);
34   border-radius: 4px;
35 }
36
37 .select2-container--admin-autocomplete .select2-selection--single .select2-selection__rendered {
38   color: var(--body-fg);
39   line-height: 30px;
40 }
41
42 .select2-container--admin-autocomplete .select2-selection--single .select2-selection__clear {
43   cursor: pointer;
44   float: right;
45   font-weight: bold;
46 }
47
48 .select2-container--admin-autocomplete .select2-selection--single .select2-selection__placeholder {
49   color: var(--body-quiet-color);
50 }
51
52 .select2-container--admin-autocomplete .select2-selection--single .select2-selection__arrow {
53   height: 26px;
54   position: absolute;
55   top: 1px;
56   right: 1px;
57   width: 20px;
58 }
59
60 .select2-container--admin-autocomplete .select2-selection--single .select2-selection__arrow b {
61   border-color: #888 transparent transparent transparent;
62   border-style: solid;
63   border-width: 5px 4px 0 4px;
64   height: 0;
65   left: 50%;
66   margin-left: -4px;
67   margin-top: -2px;
68   position: absolute;
69   top: 50%;
70   width: 0;
71 }
72
73 .select2-container--admin-autocomplete[dir="rtl"] .select2-selection--single .select2-selection__clear {
74   float: left;
75 }
76
77 .select2-container--admin-autocomplete[dir="rtl"] .select2-selection--single .select2-selection__arrow {
78   left: 1px;
79   right: auto;
80 }
81
82 .select2-container--admin-autocomplete.select2-container--disabled .select2-selection--single {
83   background-color: var(--darkened-bg);

```

```

84     cursor: default;
85 }
86
87 .select2-container--admin-autocomplete.select2-container--disabled .select2-selection--single .select2-selection__clear {
88     display: none;
89 }
90
91 .select2-container--admin-autocomplete.select2-container--open .select2-selection--single .select2-selection__arrow b {
92     border-color: transparent transparent #888 transparent;
93     border-width: 0 4px 5px 4px;
94 }
95
96 .select2-container--admin-autocomplete .select2-selection--multiple {
97     background-color: var(--body-bg);
98     border: 1px solid var(--border-color);
99     border-radius: 4px;
100     cursor: text;
101 }
102
103 .select2-container--admin-autocomplete .select2-selection--multiple .select2-selection__rendered {
104     box-sizing: border-box;
105     list-style: none;
106     margin: 0;
107     padding: 0 10px 5px 5px;
108     width: 100%;
109     display: flex;
110     flex-wrap: wrap;
111 }
112
113 .select2-container--admin-autocomplete .select2-selection--multiple .select2-selection__rendered li {
114     list-style: none;
115 }
116
117 .select2-container--admin-autocomplete .select2-selection--multiple .select2-selection__placeholder {
118     color: var(--body-quiet-color);
119     margin-top: 5px;
120     float: left;
121 }
122
123 .select2-container--admin-autocomplete .select2-selection--multiple .select2-selection__clear {
124     cursor: pointer;
125     float: right;
126     font-weight: bold;
127     margin: 5px;
128     position: absolute;
129     right: 0;
130 }
131
132 .select2-container--admin-autocomplete .select2-selection--multiple .select2-selection__choice {
133     background-color: var(--darkened-bg);
134     border: 1px solid var(--border-color);
135     border-radius: 4px;
136     cursor: default;
137     float: left;
138     margin-right: 5px;
139     margin-top: 5px;
140     padding: 0 5px;
141 }
142
143 .select2-container--admin-autocomplete .select2-selection--multiple .select2-selection__choice__remove {
144     color: var(--body-quiet-color);
145     cursor: pointer;
146     display: inline-block;
147     font-weight: bold;
148     margin-right: 2px;
149 }
150
151 .select2-container--admin-autocomplete .select2-selection--multiple .select2-selection__choice__remove:hover {
152     color: var(--body-fg);
153 }
154
155 .select2-container--admin-autocomplete[dir="rtl"] .select2-selection--multiple .select2-selection__choice, .select2-container--admin-autocomplete[dir="rtl"] .select2-selection--multiple .select2-s
156     float: right;
157 }
158
159 .select2-container--admin-autocomplete[dir="rtl"] .select2-selection--multiple .select2-selection__choice {
160     margin-left: 5px;
161     margin-right: auto;
162 }
163
164 .select2-container--admin-autocomplete[dir="rtl"] .select2-selection--multiple .select2-selection__choice__remove {
165     margin-left: 2px;
166     margin-right: auto;
167 }
168
169 .select2-container--admin-autocomplete.select2-container--focus .select2-selection--multiple {
170     border: solid var(--body-quiet-color) 1px;
171     outline: 0;
172 }
173
174 .select2-container--admin-autocomplete.select2-container--disabled .select2-selection--multiple {
175     background-color: var(--darkened-bg);
176     cursor: default;
177 }
178
179 .select2-container--admin-autocomplete.select2-container--disabled .select2-selection__choice__remove {
180     display: none;
181 }
182
183 .select2-container--admin-autocomplete.select2-container--open.select2-container--above .select2-selection--single, .select2-container--admin-autocomplete.select2-container--open.select2-con
184     border-top-left-radius: 0;
185     border-top-right-radius: 0;
186 }
187
188 .select2-container--admin-autocomplete.select2-container--open.select2-container--below .select2-selection--single, .select2-container--admin-autocomplete.select2-container--open.select2-con
189     border-bottom-left-radius: 0;
190     border-bottom-right-radius: 0;
191 }
192
193 .select2-container--admin-autocomplete .select2-search--dropdown {
194     background: var(--darkened-bg);
195 }
196
197 .select2-container--admin-autocomplete .select2-search--dropdown .select2-search__field {
198     background: var(--body-bg);

```

```

199 color: var(--body-fg);
200 border: 1px solid var(--border-color);
201 border-radius: 4px;
202 }
203
204 select2-container--admin-autocomplete .select2-search--inline .select2-search__field {
205   background: transparent;
206   color: var(--body-fg);
207   border: none;
208   outline: 0;
209   box-shadow: none;
210   -webkit-appearance: textfield;
211 }
212
213 select2-container--admin-autocomplete .select2-results > .select2-results__options {
214   max-height: 200px;
215   overflow-y: auto;
216   color: var(--body-fg);
217   background: var(--body-bg);
218 }
219
220 select2-container--admin-autocomplete .select2-results__option[role=group] {
221   padding: 0;
222 }
223
224 select2-container--admin-autocomplete .select2-results__option[aria-disabled=true] {
225   color: var(--body-quiet-color);
226 }
227
228 select2-container--admin-autocomplete .select2-results__option[aria-selected=true] {
229   background-color: var(--selected-bg);
230   color: var(--body-fg);
231 }
232
233 select2-container--admin-autocomplete .select2-results__option .select2-results__option {
234   padding-left: 1em;
235 }
236
237 select2-container--admin-autocomplete .select2-results__option .select2-results__option .select2-results__group {
238   padding-left: 0;
239 }
240
241 select2-container--admin-autocomplete .select2-results__option .select2-results__option .select2-results__option {
242   margin-left: -1em;
243   padding-left: 2em;
244 }
245
246 select2-container--admin-autocomplete .select2-results__option .select2-results__option .select2-results__option .select2-results__option {
247   margin-left: -2em;
248   padding-left: 3em;
249 }
250
251 select2-container--admin-autocomplete .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2-results__option {
252   margin-left: -3em;
253   padding-left: 4em;
254 }
255
256 select2-container--admin-autocomplete .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2-results__option {
257   margin-left: -4em;
258   padding-left: 5em;
259 }
260
261 select2-container--admin-autocomplete .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2
262   margin-left: -5em;
263   padding-left: 6em;
264 }
265
266 select2-container--admin-autocomplete .select2-results__option--highlighted[aria-selected] {
267   background-color: var(--primary);
268   color: var(--primary-fg);
269 }
270
271 select2-container--admin-autocomplete .select2-results__group {
272   cursor: default;
273   display: block;
274   padding: 6px;
275 }

```

#### File: ./static/admin/css/fonts.css

```

1  @font-face {
2    font-family: 'Roboto';
3    src: url("../fonts/Roboto-Bold-webfont.woff");
4    font-weight: 700;
5    font-style: normal;
6  }
7
8  @font-face {
9    font-family: 'Roboto';
10   src: url("../fonts/Roboto-Regular-webfont.woff");
11   font-weight: 400;
12   font-style: normal;
13 }
14
15 @font-face {
16   font-family: 'Roboto';
17   src: url("../fonts/Roboto-Light-webfont.woff");
18   font-weight: 300;
19   font-style: normal;
20 }

```

#### File: ./static/admin/css/responsive.css

```

1  /* Tablets */
2
3  input[type="submit"], button {

```



```

4     -webkit-appearance: none;
5     appearance: none;
6 }
7
8 @media (max-width: 1024px) {
9     /* Basic */
10
11     html {
12         -webkit-text-size-adjust: 100%;
13     }
14
15     td, th {
16         padding: 10px;
17         font-size: 14px;
18     }
19
20     .small {
21         font-size: 12px;
22     }
23
24     /* Layout */
25
26     #container {
27         min-width: 0;
28     }
29
30     #content {
31         padding: 20px 30px 30px;
32     }
33
34     div.breadcrumbs {
35         padding: 10px 30px;
36     }
37
38     /* Header */
39
40     #header {
41         flex-direction: column;
42         padding: 15px 30px;
43         justify-content: flex-start;
44     }
45
46     #branding h1 {
47         margin: 0 0 8px;
48         font-size: 20px;
49         line-height: 1.2;
50     }
51
52     #user-tools {
53         margin: 0;
54         font-weight: 400;
55         line-height: 1.85;
56         text-align: left;
57     }
58
59     #user-tools a {
60         display: inline-block;
61         line-height: 1.4;
62     }
63
64     /* Dashboard */
65
66     .dashboard #content {
67         width: auto;
68     }
69
70     #content-related {
71         margin-right: -290px;
72     }
73
74     .colSM #content-related {
75         margin-left: -290px;
76     }
77
78     .colMS {
79         margin-right: 290px;
80     }
81
82     .colSM {
83         margin-left: 290px;
84     }
85
86     .dashboard .module table td a {
87         padding-right: 0;
88     }
89
90     td .changelink, td .addlink {
91         font-size: 13px;
92     }
93
94     /* Changelist */
95
96     #toolbar {
97         border: none;
98         padding: 15px;
99     }
100
101     #changelist-search > div {
102         display: flex;
103         flex-wrap: nowrap;
104         max-width: 480px;
105     }
106
107     #changelist-search label {
108         line-height: 22px;
109     }
110
111     #toolbar form #searchbar {
112         flex: 1 0 auto;
113         width: 0;
114         height: 22px;
115         margin: 0 10px 0 6px;
116     }
117
118     #toolbar form input[type=submit] {

```

```

119     flex: 0 1 auto;
120 }
121
122 #changelist-search .quiet {
123     width: 0;
124     flex: 1 0 auto;
125     margin: 5px 0 0 25px;
126 }
127
128 #changelist .actions {
129     display: flex;
130     flex-wrap: wrap;
131     padding: 15px 0;
132 }
133
134 #changelist .actions.selected {
135     border: none;
136 }
137
138 #changelist .actions label {
139     display: flex;
140 }
141
142 #changelist .actions select {
143     background: var(--body-bg);
144 }
145
146 #changelist .actions .button {
147     min-width: 48px;
148     margin: 0 10px;
149 }
150
151 #changelist .actions span.all,
152 #changelist .actions span.clear,
153 #changelist .actions span.question,
154 #changelist .actions span.action-counter {
155     font-size: 11px;
156     margin: 0 10px 0 0;
157 }
158
159 #changelist-filter {
160     flex-basis: 200px;
161 }
162
163 .change-list .filtered .results,
164 .change-list .filtered .paginator,
165 .filtered #toolbar,
166 .filtered .actions,
167
168 #changelist .paginator {
169     border-top-color: var(--hairline-color); /* XXX Is this used at all? */
170 }
171
172 #changelist .results + .paginator {
173     border-top: none;
174 }
175
176 /* Forms */
177
178 label {
179     font-size: 14px;
180 }
181
182 .form-row input[type=text],
183 .form-row input[type=password],
184 .form-row input[type=email],
185 .form-row input[type=url],
186 .form-row input[type=tel],
187 .form-row input[type=number],
188 .form-row textarea,
189 .form-row select,
190 .form-row .vTextField {
191     box-sizing: border-box;
192     margin: 0;
193     padding: 6px 8px;
194     min-height: 36px;
195     font-size: 14px;
196 }
197
198 .form-row select {
199     height: 36px;
200 }
201
202 .form-row select[multiple] {
203     height: auto;
204     min-height: 0;
205 }
206
207 fieldset .fieldBox {
208     float: none;
209     margin: 0 -10px;
210     padding: 0 10px;
211 }
212
213 fieldset .fieldBox + .fieldBox {
214     margin-top: 10px;
215     padding-top: 10px;
216     border-top: 1px solid var(--hairline-color);
217 }
218
219 textarea {
220     max-width: 100%;
221     max-height: 120px;
222 }
223
224 .aligned label {
225     padding-top: 6px;
226 }
227
228 .aligned .related-lookup,
229 .aligned .datetimeshortcuts,
230 .aligned .related-lookup + strong {
231     align-self: center;
232     margin-left: 15px;
233 }

```

```

234
235 form .aligned ul.radiolist {
236     margin-left: 2px;
237 }
238
239 /* Related widget */
240
241 .related-widget-wrapper {
242     float: none;
243 }
244
245 .related-widget-wrapper-link + .selector {
246     max-width: calc(100% - 30px);
247     margin-right: 15px;
248 }
249
250 select + .related-widget-wrapper-link,
251 .related-widget-wrapper-link + .related-widget-wrapper-link {
252     margin-left: 10px;
253 }
254
255 /* Selector */
256
257 .selector {
258     display: flex;
259     width: 100%;
260 }
261
262 .selector .selector-filter {
263     display: flex;
264     align-items: center;
265 }
266
267 .selector .selector-filter label {
268     margin: 0 8px 0 0;
269 }
270
271 .selector .selector-filter input {
272     width: auto;
273     min-height: 0;
274     flex: 1 1;
275 }
276
277 .selector-available, .selector-chosen {
278     width: auto;
279     flex: 1 1;
280     display: flex;
281     flex-direction: column;
282 }
283
284 .selector select {
285     width: 100%;
286     flex: 1 0 auto;
287     margin-bottom: 5px;
288 }
289
290 .selector ul.selector-chooser {
291     width: 26px;
292     height: 52px;
293     padding: 2px 0;
294     margin: auto 15px;
295     border-radius: 20px;
296     transform: translateY(-10px);
297 }
298
299 .selector-add, .selector-remove {
300     width: 20px;
301     height: 20px;
302     background-size: 20px auto;
303 }
304
305 .selector-add {
306     background-position: 0 -120px;
307 }
308
309 .selector-remove {
310     background-position: 0 -80px;
311 }
312
313 a.selector-chooseall, a.selector-clearall {
314     align-self: center;
315 }
316
317 .stacked {
318     flex-direction: column;
319     max-width: 480px;
320 }
321
322 .stacked > * {
323     flex: 0 1 auto;
324 }
325
326 .stacked select {
327     margin-bottom: 0;
328 }
329
330 .stacked .selector-available, .stacked .selector-chosen {
331     width: auto;
332 }
333
334 .stacked ul.selector-chooser {
335     width: 52px;
336     height: 26px;
337     padding: 0 2px;
338     margin: 15px auto;
339     transform: none;
340 }
341
342 .stacked .selector-chooser li {
343     padding: 3px;
344 }
345
346 .stacked .selector-add, .stacked .selector-remove {
347     background-size: 20px auto;
348 }

```

```

349
350 .stacked.selector-add {
351     background-position: 0 -40px;
352 }
353
354 .stacked.active.selector-add {
355     background-position: 0 -40px;
356 }
357
358 .active.selector-add:focus, .active.selector-add:hover {
359     background-position: 0 -140px;
360 }
361
362 .stacked.active.selector-add:focus, .stacked.active.selector-add:hover {
363     background-position: 0 -60px;
364 }
365
366 .stacked.selector-remove {
367     background-position: 0 0;
368 }
369
370 .stacked.active.selector-remove {
371     background-position: 0 0;
372 }
373
374 .active.selector-remove:focus, .active.selector-remove:hover {
375     background-position: 0 -100px;
376 }
377
378 .stacked.active.selector-remove:focus, .stacked.active.selector-remove:hover {
379     background-position: 0 -20px;
380 }
381
382 .help-tooltip, .selector.help-icon {
383     display: none;
384 }
385
386 form, form-row p.datetime {
387     width: 100%;
388 }
389
390 .datetime input {
391     width: 50%;
392     max-width: 120px;
393 }
394
395 .datetime span {
396     font-size: 13px;
397 }
398
399 .datetime.timezonewarning {
400     display: block;
401     font-size: 11px;
402     color: var(--body-quiet-color);
403 }
404
405 .datetimeshortcuts {
406     color: var(--border-color); /* XXX Redundant, .datetime span also sets #ccc */
407 }
408
409 .form-row.datetime input.vDateField, .form-row.datetime input.vTimeField {
410     width: 75%;
411 }
412
413 .inline-group {
414     overflow: auto;
415 }
416
417 /* Messages */
418
419 ul.messagelist li {
420     padding-left: 55px;
421     background-position: 30px 12px;
422 }
423
424 ul.messagelist li.error {
425     background-position: 30px 12px;
426 }
427
428 ul.messagelist li.warning {
429     background-position: 30px 14px;
430 }
431
432 /* Login */
433
434 .login #header {
435     padding: 15px 20px;
436 }
437
438 .login #branding h1 {
439     margin: 0;
440 }
441
442 /* GIS */
443
444 div.olMap {
445     max-width: calc(100vw - 30px);
446     max-height: 300px;
447 }
448
449 .olMap + .clear.features {
450     display: block;
451     margin-top: 10px;
452 }
453
454 /* Docs */
455
456 .module table.xfull {
457     width: 100%;
458 }
459
460 pre.literal-block {
461     overflow: auto;
462 }
463

```

```

464
465 /* Mobile */
466
467 @media (max-width: 767px) {
468 /* Layout */
469
470 #header, #content, #footer {
471 padding: 15px;
472 }
473
474 #footer:empty {
475 padding: 0;
476 }
477
478 div.breadcrumbs {
479 padding: 10px 15px;
480 }
481
482 /* Dashboard */
483
484 .colMS, .colSM {
485 margin: 0;
486 }
487
488 #content-related, .colSM #content-related {
489 width: 100%;
490 margin: 0;
491 }
492
493 #content-related .module {
494 margin-bottom: 0;
495 }
496
497 #content-related .module h2 {
498 padding: 10px 15px;
499 font-size: 16px;
500 }
501
502 /* Changelist */
503
504 #changelist {
505 align-items: stretch;
506 flex-direction: column;
507 }
508
509 #toolbar {
510 padding: 10px;
511 }
512
513 #changelist-filter {
514 margin-left: 0;
515 }
516
517 #changelist .actions label {
518 flex: 1 1;
519 }
520
521 #changelist .actions select {
522 flex: 1 0;
523 width: 100%;
524 }
525
526 #changelist .actions span {
527 flex: 1 0 100%;
528 }
529
530 #changelist-filter {
531 position: static;
532 width: auto;
533 margin-top: 30px;
534 }
535
536 .object-tools {
537 float: none;
538 margin: 0 0 15px;
539 padding: 0;
540 overflow: hidden;
541 }
542
543 .object-tools li {
544 height: auto;
545 margin-left: 0;
546 }
547
548 .object-tools li + li {
549 margin-left: 15px;
550 }
551
552 /* Forms */
553
554 .form-row {
555 padding: 15px 0;
556 }
557
558 .aligned .form-row,
559 .aligned .form-row > div {
560 display: flex;
561 flex-wrap: wrap;
562 max-width: 100vw;
563 }
564
565 .aligned .form-row > div {
566 width: calc(100vw - 30px);
567 }
568
569 textarea {
570 max-width: none;
571 }
572
573 .vURLField {
574 width: auto;
575 }
576
577 fieldset .fieldBox + .fieldBox {
578 margin-top: 15px;

```

```

579     padding-top: 15px;
580 }
581
582 fieldset.collapsed .form-row {
583     display: none;
584 }
585
586 .aligned label {
587     width: 100%;
588     padding: 0 0 10px;
589 }
590
591 .aligned label:after {
592     max-height: 0;
593 }
594
595 .aligned .form-row input,
596 .aligned .form-row select,
597 .aligned .form-row textarea {
598     flex: 1 1 auto;
599     max-width: 100%;
600 }
601
602 .aligned .checkbox-row {
603     align-items: center;
604 }
605
606 .aligned .checkbox-row input {
607     flex: 0 1 auto;
608     margin: 0;
609 }
610
611 .aligned .vCheckboxLabel {
612     flex: 1 0;
613     padding: 1px 0 0 5px;
614 }
615
616 .aligned label + p,
617 .aligned label + div.help,
618 .aligned label + div.readonly {
619     padding: 0;
620     margin-left: 0;
621 }
622
623 .aligned p.file-upload {
624     margin-left: 0;
625     font-size: 13px;
626 }
627
628 span.clearable-file-input {
629     margin-left: 15px;
630 }
631
632 span.clearable-file-input label {
633     font-size: 13px;
634     padding-bottom: 0;
635 }
636
637 .aligned .timezonewarning {
638     flex: 1 0 100%;
639     margin-top: 5px;
640 }
641
642 form .aligned .form-row div.help {
643     width: 100%;
644     margin: 5px 0 0;
645     padding: 0;
646 }
647
648 form .aligned ul {
649     margin-left: 0;
650     padding-left: 0;
651 }
652
653 form .aligned ul.radiolist {
654     margin-right: 15px;
655     margin-bottom: -3px;
656 }
657
658 form .aligned ul.radiolist:not(.inline) li + li {
659     margin-top: 5px;
660 }
661
662 /* Related widget */
663
664 .related-widget-wrapper {
665     width: 100%;
666     display: flex;
667     align-items: flex-start;
668 }
669
670 .related-widget-wrapper .selector {
671     order: 1;
672 }
673
674 .related-widget-wrapper > a {
675     order: 2;
676 }
677
678 .related-widget-wrapper .radiolist ~ a {
679     align-self: flex-end;
680 }
681
682 .related-widget-wrapper > select ~ a {
683     align-self: center;
684 }
685
686 select + .related-widget-wrapper-link,
687 .related-widget-wrapper-link + .related-widget-wrapper-link {
688     margin-left: 15px;
689 }
690
691 /* Selector */
692
693 .selector {

```

```

694     flex-direction: column;
695 }
696
697 .selector > * {
698     float: none;
699 }
700
701 .selector-available, .selector-chosen {
702     margin-bottom: 0;
703     flex: 1 1 auto;
704 }
705
706 .selector select {
707     max-height: 96px;
708 }
709
710 .selector ul.selector-chooser {
711     display: block;
712     float: none;
713     width: 52px;
714     height: 26px;
715     padding: 0 2px;
716     margin: 15px auto 20px;
717     transform: none;
718 }
719
720 .selector ul.selector-chooser li {
721     float: left;
722 }
723
724 .selector-remove {
725     background-position: 0 0;
726 }
727
728 .active.selector-remove:focus, .active.selector-remove:hover {
729     background-position: 0 -20px;
730 }
731
732 .selector-add {
733     background-position: 0 -40px;
734 }
735
736 .active.selector-add:focus, .active.selector-add:hover {
737     background-position: 0 -60px;
738 }
739
740 /* Inlines */
741
742 .inline-group[data-inline-type="stacked"].inline-related {
743     border: 1px solid var(--hairline-color);
744     border-radius: 4px;
745     margin-top: 15px;
746     overflow: auto;
747 }
748
749 .inline-group[data-inline-type="stacked"].inline-related > * {
750     box-sizing: border-box;
751 }
752
753 .inline-group[data-inline-type="stacked"].inline-related .module {
754     padding: 0 10px;
755 }
756
757 .inline-group[data-inline-type="stacked"].inline-related .module .form-row {
758     border-top: 1px solid var(--hairline-color);
759     border-bottom: none;
760 }
761
762 .inline-group[data-inline-type="stacked"].inline-related .module .form-row:first-child {
763     border-top: none;
764 }
765
766 .inline-group[data-inline-type="stacked"].inline-related h3 {
767     padding: 10px;
768     border-top-width: 0;
769     border-bottom-width: 2px;
770     display: flex;
771     flex-wrap: wrap;
772     align-items: center;
773 }
774
775 .inline-group[data-inline-type="stacked"].inline-related h3 .inline_label {
776     margin-right: auto;
777 }
778
779 .inline-group[data-inline-type="stacked"].inline-related h3 span.delete {
780     float: none;
781     flex: 1 1 100%;
782     margin-top: 5px;
783 }
784
785 .inline-group[data-inline-type="stacked"].aligned .form-row > div:not([class]) {
786     width: 100%;
787 }
788
789 .inline-group[data-inline-type="stacked"].aligned label {
790     width: 100%;
791 }
792
793 .inline-group[data-inline-type="stacked"] div.add-row {
794     margin-top: 15px;
795     border: 1px solid var(--hairline-color);
796     border-radius: 4px;
797 }
798
799 .inline-group div.add-row,
800 .inline-group .tabular tr.add-row td {
801     padding: 0;
802 }
803
804 .inline-group div.add-row a,
805 .inline-group .tabular tr.add-row td a {
806     display: block;
807     padding: 8px 10px 8px 26px;
808     background-position: 8px 9px;

```

```

809 }
810
811 /* Submit row */
812
813 .submit-row {
814     padding: 10px 10px 0;
815     margin: 0 0 15px;
816     display: flex;
817     flex-direction: column;
818 }
819
820 .submit-row > * {
821     width: 100%;
822 }
823
824 .submit-row input, .submit-row input.default, .submit-row a, .submit-row a.closetlink {
825     float: none;
826     margin: 0 0 10px;
827     text-align: center;
828 }
829
830 .submit-row a.closetlink {
831     padding: 10px 0;
832 }
833
834 .submit-row p.deletelink-box {
835     order: 4;
836 }
837
838 /* Messages */
839
840 ul.messagelist li {
841     padding-left: 40px;
842     background-position: 15px 12px;
843 }
844
845 ul.messagelist li.error {
846     background-position: 15px 12px;
847 }
848
849 ul.messagelist li.warning {
850     background-position: 15px 14px;
851 }
852
853 /* Paginator */
854
855 .paginator .this-page, .paginator a:link, .paginator a:visited {
856     padding: 4px 10px;
857 }
858
859 /* Login */
860
861 body.login {
862     padding: 0 15px;
863 }
864
865 .login #container {
866     width: auto;
867     max-width: 480px;
868     margin: 50px auto;
869 }
870
871 .login #header,
872 .login #content {
873     padding: 15px;
874 }
875
876 .login #content-main {
877     float: none;
878 }
879
880 .login .form-row {
881     padding: 0;
882 }
883
884 .login .form-row + .form-row {
885     margin-top: 15px;
886 }
887
888 .login .form-row label {
889     margin: 0 0 5px;
890     line-height: 1.2;
891 }
892
893 .login .submit-row {
894     padding: 15px 0 0;
895 }
896
897 .login br {
898     display: none;
899 }
900
901 .login .submit-row input {
902     margin: 0;
903     text-transform: uppercase;
904 }
905
906 .errornote {
907     margin: 0 0 20px;
908     padding: 8px 12px;
909     font-size: 13px;
910 }
911
912 /* Calendar and clock */
913
914 .calendarbox, .clockbox {
915     position: fixed !important;
916     top: 50% !important;
917     left: 50% !important;
918     transform: translate(-50%, -50%);
919     margin: 0;
920     border: none;
921     overflow: visible;
922 }
923

```



```

924 .calendarbox:before, .clockbox:before {
925     content: "";
926     position: fixed;
927     top: 50%;
928     left: 50%;
929     width: 100vw;
930     height: 100vh;
931     background: rgba(0, 0, 0, 0.75);
932     transform: translate(-50%, -50%);
933 }
934
935 .calendarbox > *, .clockbox > * {
936     position: relative;
937     z-index: 1;
938 }
939
940 .calendarbox > div:first-child {
941     z-index: 2;
942 }
943
944 .calendarbox .calendar, .clockbox h2 {
945     border-radius: 4px 4px 0 0;
946     overflow: hidden;
947 }
948
949 .calendarbox .calendar-cancel, .clockbox .calendar-cancel {
950     border-radius: 0 0 4px 4px;
951     overflow: hidden;
952 }
953
954 .calendar-shortcuts {
955     padding: 10px 0;
956     font-size: 12px;
957     line-height: 12px;
958 }
959
960 .calendar-shortcuts a {
961     margin: 0 4px;
962 }
963
964 .timelist a {
965     background: var(--body-bg);
966     padding: 4px;
967 }
968
969 .calendar-cancel {
970     padding: 8px 10px;
971 }
972
973 .clockbox h2 {
974     padding: 8px 15px;
975 }
976
977 .calendar-caption {
978     padding: 10px;
979 }
980
981 .calendarbox .calendarnav-previous, .calendarbox .calendarnav-next {
982     z-index: 1;
983     top: 10px;
984 }
985
986 /* History */
987
988 table#change-history tbody th, table#change-history tbody td {
989     font-size: 13px;
990     word-break: break-word;
991 }
992
993 table#change-history tbody th {
994     width: auto;
995 }
996
997 /* Docs */
998
999 table.model tbody th, table.model tbody td {
1000     font-size: 13px;
1001     word-break: break-word;
1002 }
1003 }

```

#### File: ./static/admin/css/vendor/select2/select2.min.css

```

1 .select2-container { box-sizing: border-box; display: inline-block; margin: 0; position: relative; vertical-align: middle; }
2 .select2-container .select2-selection--single { box-sizing: border-box; cursor: pointer; display: inline-block;

```

#### File: ./static/admin/css/vendor/select2/select2.css

```

1 .select2-container {
2     box-sizing: border-box;
3     display: inline-block;
4     margin: 0;
5     position: relative;
6     vertical-align: middle; }
7 .select2-container .select2-selection--single {
8     box-sizing: border-box;
9     cursor: pointer;
10    display: block;
11    height: 28px;
12    user-select: none;
13    -webkit-user-select: none; }

```

```

14 .select2-container .select2-selection--single .select2-selection__rendered {
15     display: block;
16     padding-left: 8px;
17     padding-right: 20px;
18     overflow: hidden;
19     text-overflow: ellipsis;
20     white-space: nowrap; }
21 .select2-container .select2-selection--single .select2-selection__clear {
22     position: relative; }
23 .select2-container[dir="rtl"] .select2-selection--single .select2-selection__rendered {
24     padding-right: 8px;
25     padding-left: 20px; }
26 .select2-container .select2-selection--multiple {
27     box-sizing: border-box;
28     cursor: pointer;
29     display: block;
30     min-height: 32px;
31     user-select: none;
32     -webkit-user-select: none; }
33 .select2-container .select2-selection--multiple .select2-selection__rendered {
34     display: inline-block;
35     overflow: hidden;
36     padding-left: 8px;
37     text-overflow: ellipsis;
38     white-space: nowrap; }
39 .select2-container .select2-search--inline {
40     float: left; }
41 .select2-container .select2-search--inline .select2-search__field {
42     box-sizing: border-box;
43     border: none;
44     font-size: 100%;
45     margin-top: 5px;
46     padding: 0; }
47 .select2-container .select2-search--inline .select2-search__field::-webkit-search-cancel-button {
48     -webkit-appearance: none; }
49
50 .select2-dropdown {
51     background-color: white;
52     border: 1px solid #aaa;
53     border-radius: 4px;
54     box-sizing: border-box;
55     display: block;
56     position: absolute;
57     left: -10000px;
58     width: 100%;
59     z-index: 1051; }
60
61 .select2-results {
62     display: block; }
63
64 .select2-results__options {
65     list-style: none;
66     margin: 0;
67     padding: 0; }
68
69 .select2-results__option {
70     padding: 6px;
71     user-select: none;
72     -webkit-user-select: none; }
73 .select2-results__option[aria-selected] {
74     cursor: pointer; }
75
76 .select2-container--open .select2-dropdown {
77     left: 0; }
78
79 .select2-container--open .select2-dropdown--above {
80     border-bottom: none;
81     border-bottom-left-radius: 0;
82     border-bottom-right-radius: 0; }
83
84 .select2-container--open .select2-dropdown--below {
85     border-top: none;
86     border-top-left-radius: 0;
87     border-top-right-radius: 0; }
88
89 .select2-search--dropdown {
90     display: block;
91     padding: 4px; }
92 .select2-search--dropdown .select2-search__field {
93     padding: 4px;
94     width: 100%;
95     box-sizing: border-box; }
96 .select2-search--dropdown .select2-search__field::-webkit-search-cancel-button {
97     -webkit-appearance: none; }
98 .select2-search--dropdown .select2-search--hide {
99     display: none; }
100
101 .select2-close-mask {
102     border: 0;
103     margin: 0;
104     padding: 0;
105     display: block;
106     position: fixed;
107     left: 0;
108     top: 0;
109     min-height: 100%;
110     min-width: 100%;
111     height: auto;
112     width: auto;
113     opacity: 0;
114     z-index: 99;
115     background-color: #fff;
116     filter: alpha(opacity=0); }
117
118 .select2-hidden-accessible {
119     border: 0 !important;
120     clip: rect(0 0 0 0) !important;
121     -webkit-clip-path: inset(50%) !important;
122     clip-path: inset(50%) !important;
123     height: 1px !important;
124     overflow: hidden !important;
125     padding: 0 !important;
126     position: absolute !important;
127     width: 1px !important;
128     white-space: nowrap !important; }

```

```
129
130 .select2-container--default .select2-selection--single {
131   background-color: #fff;
132   border: 1px solid #aaa;
133   border-radius: 4px; }
134 .select2-container--default .select2-selection--single .select2-selection__rendered {
135   color: #444;
136   line-height: 28px; }
137 .select2-container--default .select2-selection--single .select2-selection__clear {
138   cursor: pointer;
139   float: right;
140   font-weight: bold; }
141 .select2-container--default .select2-selection--single .select2-selection__placeholder {
142   color: #999; }
143 .select2-container--default .select2-selection--single .select2-selection__arrow {
144   height: 26px;
145   position: absolute;
146   top: 1px;
147   right: 1px;
148   width: 20px; }
149 .select2-container--default .select2-selection--single .select2-selection__arrow b {
150   border-color: #888 transparent transparent;
151   border-style: solid;
152   border-width: 5px 4px 0 4px;
153   height: 0;
154   left: 50%;
155   margin-left: -4px;
156   margin-top: -2px;
157   position: absolute;
158   top: 50%;
159   width: 0; }
160
161 .select2-container--default[dir="rtl"] .select2-selection--single .select2-selection__clear {
162   float: left; }
163
164 .select2-container--default[dir="rtl"] .select2-selection--single .select2-selection__arrow {
165   left: 1px;
166   right: auto; }
167
168 .select2-container--default.select2-container--disabled .select2-selection--single {
169   background-color: #eee;
170   cursor: default; }
171 .select2-container--default.select2-container--disabled .select2-selection--single .select2-selection__clear {
172   display: none; }
173
174 .select2-container--default.select2-container--open .select2-selection--single .select2-selection__arrow b {
175   border-color: transparent transparent #888 transparent;
176   border-width: 0 4px 5px 4px; }
177
178 .select2-container--default .select2-selection--multiple {
179   background-color: white;
180   border: 1px solid #aaa;
181   border-radius: 4px;
182   cursor: text; }
183 .select2-container--default .select2-selection--multiple .select2-selection__rendered {
184   box-sizing: border-box;
185   list-style: none;
186   margin: 0;
187   padding: 0 5px;
188   width: 100%; }
189 .select2-container--default .select2-selection--multiple .select2-selection__rendered li {
190   list-style: none; }
191 .select2-container--default .select2-selection--multiple .select2-selection__clear {
192   cursor: pointer;
193   float: right;
194   font-weight: bold;
195   margin-top: 5px;
196   margin-right: 10px;
197   padding: 1px; }
198 .select2-container--default .select2-selection--multiple .select2-selection__choice {
199   background-color: #e4e4e4;
200   border: 1px solid #aaa;
201   border-radius: 4px;
202   cursor: default;
203   float: left;
204   margin-right: 5px;
205   margin-top: 5px;
206   padding: 0 5px; }
207 .select2-container--default .select2-selection--multiple .select2-selection__choice__remove {
208   color: #999;
209   cursor: pointer;
210   display: inline-block;
211   font-weight: bold;
212   margin-right: 2px; }
213 .select2-container--default .select2-selection--multiple .select2-selection__choice__remove:hover {
214   color: #333; }
215
216 .select2-container--default[dir="rtl"] .select2-selection--multiple .select2-selection__choice, .select2-container--default[dir="rtl"] .select2-selection--multiple .select2-search--inline {
217   float: right; }
218
219 .select2-container--default[dir="rtl"] .select2-selection--multiple .select2-selection__choice {
220   margin-left: 5px;
221   margin-right: auto; }
222
223 .select2-container--default[dir="rtl"] .select2-selection--multiple .select2-selection__choice__remove {
224   margin-left: 2px;
225   margin-right: auto; }
226
227 .select2-container--default.select2-container--focus .select2-selection--multiple {
228   border: solid black 1px;
229   outline: 0; }
230
231 .select2-container--default.select2-container--disabled .select2-selection--multiple {
232   background-color: #eee;
233   cursor: default; }
234
235 .select2-container--default.select2-container--disabled .select2-selection__choice__remove {
236   display: none; }
237
238 .select2-container--default.select2-container--open.select2-container--above .select2-selection--single, .select2-container--default.select2-container--open.select2-container--above .select2-seler
239   border-top-left-radius: 0;
240   border-top-right-radius: 0; }
241
242 .select2-container--default.select2-container--open.select2-container--below .select2-selection--single, .select2-container--default.select2-container--open.select2-container--below .select2-selec
243   border-bottom-left-radius: 0;
```

```
244 border-bottom-right-radius: 0; }
245
246 select2-container--default .select2-search--dropdown .select2-search__field {
247   border: 1px solid #aaa; }
248
249 select2-container--default .select2-search--inline .select2-search__field {
250   background: transparent;
251   border: none;
252   outline: 0;
253   box-shadow: none;
254   -webkit-appearance: textfield; }
255
256 select2-container--default .select2-results > .select2-results__options {
257   max-height: 200px;
258   overflow-y: auto; }
259
260 select2-container--default .select2-results__option[role=group] {
261   padding: 0; }
262
263 select2-container--default .select2-results__option[aria-disabled=true] {
264   color: #999; }
265
266 select2-container--default .select2-results__option[aria-selected=true] {
267   background-color: #ddd; }
268
269 select2-container--default .select2-results__option .select2-results__option {
270   padding-left: 1em; }
271 .select2-container--default .select2-results__option .select2-results__option .select2-results__group {
272   padding-left: 0; }
273 .select2-container--default .select2-results__option .select2-results__option .select2-results__option {
274   margin-left: -1em;
275   padding-left: 2em; }
276 .select2-container--default .select2-results__option .select2-results__option .select2-results__option .select2-results__option {
277   margin-left: -2em;
278   padding-left: 3em; }
279 .select2-container--default .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2-results__option {
280   margin-left: -3em;
281   padding-left: 4em; }
282 .select2-container--default .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2-results__option {
283   margin-left: -4em;
284   padding-left: 5em; }
285 .select2-container--default .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2-results__option .select2-result
286   margin-left: -5em;
287   padding-left: 6em; }
288
289 select2-container--default .select2-results__option--highlighted[aria-selected] {
290   background-color: #5897fb;
291   color: white; }
292
293 select2-container--default .select2-results__group {
294   cursor: default;
295   display: block;
296   padding: 6px; }
297
298 select2-container--classic .select2-selection--single {
299   background-color: #f7f7f7;
300   border: 1px solid #aaa;
301   border-radius: 4px;
302   outline: 0;
303   background-image: -webkit-linear-gradient(top, white 50%, #eeeeee 100%);
304   background-image: -o-linear-gradient(top, white 50%, #eeeeee 100%);
305   background-image: linear-gradient(to bottom, white 50%, #eeeeee 100%);
306   background-repeat: repeat-x;
307   filter: progid:DXImageTransform.Microsoft.gradient(startColorstr=#FFFFFF, endColorstr=#FFFFFF, GradientType=0); }
308 .select2-container--classic .select2-selection--single:focus {
309   border: 1px solid #5897fb; }
310 .select2-container--classic .select2-selection--single .select2-selection__rendered {
311   color: #444;
312   line-height: 28px; }
313 .select2-container--classic .select2-selection--single .select2-selection__clear {
314   cursor: pointer;
315   float: right;
316   font-weight: bold;
317   margin-right: 10px; }
318 .select2-container--classic .select2-selection--single .select2-selection__placeholder {
319   color: #999; }
320 .select2-container--classic .select2-selection--single .select2-selection__arrow {
321   background-color: #ddd;
322   border: none;
323   border-left: 1px solid #aaa;
324   border-top-right-radius: 4px;
325   border-bottom-right-radius: 4px;
326   height: 26px;
327   position: absolute;
328   top: 1px;
329   right: 1px;
330   width: 20px;
331   background-image: -webkit-linear-gradient(top, #eeeeee 50%, #cccccc 100%);
332   background-image: -o-linear-gradient(top, #eeeeee 50%, #cccccc 100%);
333   background-image: linear-gradient(to bottom, #eeeeee 50%, #cccccc 100%);
334   background-repeat: repeat-x;
335   filter: progid:DXImageTransform.Microsoft.gradient(startColorstr=#FFFFFF, endColorstr=#FFCCCC, GradientType=0); }
336 .select2-container--classic .select2-selection--single .select2-selection__arrow b {
337   border-color: #888 transparent transparent transparent;
338   border-style: solid;
339   border-width: 5px 4px 0 4px;
340   height: 0;
341   left: 50%;
342   margin-left: -4px;
343   margin-top: -2px;
344   position: absolute;
345   top: 50%;
346   width: 0; }
347
348 select2-container--classic[dir="rtl"] .select2-selection--single .select2-selection__clear {
349   float: left; }
350
351 select2-container--classic[dir="rtl"] .select2-selection--single .select2-selection__arrow {
352   border: none;
353   border-right: 1px solid #aaa;
354   border-radius: 0;
355   border-top-left-radius: 4px;
356   border-bottom-left-radius: 4px;
357   left: 1px;
358   right: auto; }
```

```
359
360 .select2-container--classic.select2-container--open .select2-selection--single {
361   border: 1px solid #5897fb; }
362 .select2-container--classic.select2-container--open .select2-selection--single .select2-selection__arrow {
363   background: transparent;
364   border: none; }
365 .select2-container--classic.select2-container--open .select2-selection--single .select2-selection__arrow b {
366   border-color: transparent transparent #888 transparent;
367   border-width: 0 4px 5px 4px; }
368
369 .select2-container--classic.select2-container--open.select2-container--above .select2-selection--single {
370   border-top: none;
371   border-top-left-radius: 0;
372   border-top-right-radius: 0;
373   background-image: -webkit-linear-gradient(top, white 0%, #eeeeee 50%);
374   background-image: -o-linear-gradient(top, white 0%, #eeeeee 50%);
375   background-image: linear-gradient(to bottom, white 0%, #eeeeee 50%);
376   background-repeat: repeat-x;
377   filter: progid:DXImageTransform.Microsoft.gradient(startColorstr=#FFFFFF, endColorstr=#FFFFFF, GradientType=0); }
378
379 .select2-container--classic.select2-container--open.select2-container--below .select2-selection--single {
380   border-bottom: none;
381   border-bottom-left-radius: 0;
382   border-bottom-right-radius: 0;
383   background-image: -webkit-linear-gradient(top, #eeeeee 50%, white 100%);
384   background-image: -o-linear-gradient(top, #eeeeee 50%, white 100%);
385   background-image: linear-gradient(to bottom, #eeeeee 50%, white 100%);
386   background-repeat: repeat-x;
387   filter: progid:DXImageTransform.Microsoft.gradient(startColorstr=#FFFFFF, endColorstr=#FFFFFF, GradientType=0); }
388
389 .select2-container--classic .select2-selection--multiple {
390   background-color: white;
391   border: 1px solid #aaa;
392   border-radius: 4px;
393   cursor: text;
394   outline: 0; }
395 .select2-container--classic .select2-selection--multiple:focus {
396   border: 1px solid #5897fb; }
397 .select2-container--classic .select2-selection--multiple .select2-selection__rendered {
398   list-style: none;
399   margin: 0;
400   padding: 0 5px; }
401 .select2-container--classic .select2-selection--multiple .select2-selection__clear {
402   display: none; }
403 .select2-container--classic .select2-selection--multiple .select2-selection__choice {
404   background-color: #e4e4e4;
405   border: 1px solid #aaa;
406   border-radius: 4px;
407   cursor: default;
408   float: left;
409   margin-right: 5px;
410   margin-top: 5px;
411   padding: 0 5px; }
412 .select2-container--classic .select2-selection--multiple .select2-selection__choice_remove {
413   color: #888;
414   cursor: pointer;
415   display: inline-block;
416   font-weight: bold;
417   margin-right: 2px; }
418 .select2-container--classic .select2-selection--multiple .select2-selection__choice_remove:hover {
419   color: #555; }
420
421 .select2-container--classic[dir="rtl"] .select2-selection--multiple .select2-selection__choice {
422   float: right;
423   margin-left: 5px;
424   margin-right: auto; }
425
426 .select2-container--classic[dir="rtl"] .select2-selection--multiple .select2-selection__choice_remove {
427   margin-left: 2px;
428   margin-right: auto; }
429
430 .select2-container--classic.select2-container--open .select2-selection--multiple {
431   border: 1px solid #5897fb; }
432
433 .select2-container--classic.select2-container--open.select2-container--above .select2-selection--multiple {
434   border-top: none;
435   border-top-left-radius: 0;
436   border-top-right-radius: 0; }
437
438 .select2-container--classic.select2-container--open.select2-container--below .select2-selection--multiple {
439   border-bottom: none;
440   border-bottom-left-radius: 0;
441   border-bottom-right-radius: 0; }
442
443 .select2-container--classic .select2-search--dropdown .select2-search__field {
444   border: 1px solid #aaa;
445   outline: 0; }
446
447 .select2-container--classic .select2-search--inline .select2-search__field {
448   outline: 0;
449   box-shadow: none; }
450
451 .select2-container--classic .select2-dropdown {
452   background-color: white;
453   border: 1px solid transparent; }
454
455 .select2-container--classic .select2-dropdown--above {
456   border-bottom: none; }
457
458 .select2-container--classic .select2-dropdown--below {
459   border-top: none; }
460
461 .select2-container--classic .select2-results > .select2-results__options {
462   max-height: 200px;
463   overflow-y: auto; }
464
465 .select2-container--classic .select2-results__option[role=group] {
466   padding: 0; }
467
468 .select2-container--classic .select2-results__option[aria-disabled=true] {
469   color: grey; }
470
471 .select2-container--classic .select2-results__option--highlighted[aria-selected] {
472   background-color: #3875d7;
473   color: white; }
```

```
474
475 .select2-container--classic .select2-results__group {
476   cursor: default;
477   display: block;
478   padding: 6px; }
479
480 .select2-container--classic.select2-container--open .select2-dropdown {
481   border-color: #5897fb; }
```

**File: ./static/admin/css/vendor/select2/LICENSE-SELECT2.md**

```
1 The MIT License (MIT)
2
3 Copyright (c) 2012-2017 Kevin Brown, Igor Vaynberg, and Select2 contributors
4
5 Permission is hereby granted, free of charge, to any person obtaining a copy
6 of this software and associated documentation files (the "Software"), to deal
7 in the Software without restriction, including without limitation the rights
8 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
9 copies of the Software, and to permit persons to whom the Software is
10 furnished to do so, subject to the following conditions:
11
12 The above copyright notice and this permission notice shall be included in
13 all copies or substantial portions of the Software.
14
15 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
16 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
17 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
18 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
19 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
20 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
21 THE SOFTWARE.
```

**File: ./static/admin/fonts/README.txt**

Roboto webfont source: <https://www.google.com/fonts/specimen/Roboto>  
WOFF files extracted using <https://github.com/majodev/google-webfonts-helper>  
Weights used in this project: Light (300), Regular (400), Bold (700)

**File: ./static/admin/fonts/Roboto-Bold-webfont.woff**

[binary]

**File: ./static/admin/fonts/Roboto-Light-webfont.woff**

[binary]

**File: ./static/admin/fonts/LICENSE.txt**

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

**TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION**

**1. Definitions.**

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes

of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or

implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

## END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

**File: ./static/admin/fonts/Roboto-Regular-webfont.woff**

[binary]

**File: ./static/admin/js/autocomplete.js**

```
1 'use strict';
2 {
3   const $ = django.jQuery;
4
5   $ fn djangoAdminSelect2 => function() {
6     $ each(this, function(i, element) {
7       $ element.select2({
8         ajax: {
9           data: (params) => {
10             return {
11               term: params.term,
12               page: params.page,
13               app_label: element.dataset.appLabel,
14               model_name: element.dataset.modelName,
15               field_name: element.dataset.fieldName
16             };
17           }
18         }
19       });
20     });
21     return this;
22   };
23
24   $(function() {
25     // Initialize all autocomplete widgets except the one in the template
26     // form used when a new formset is added.
27     $('admin-autocomplete').not('[name*=__prefix__]').djangoAdminSelect2();
28   });
29
30   $(document).on('formset:added', function() {
31     return function(event, $newFormset) {
32       return $newFormset.find('admin-autocomplete').djangoAdminSelect2();
33     };
34   })(this);
35 }
```

**File: ./static/admin/js/SelectFilter2.js**

```
1 /*global SelectBox, oetext, interpolate, quickElement, SelectFilter*/
```



```

2  /*
3  SelectFilter2 - Turns a multiple-select box into a filter interface.
4
5  Requires core.js and SelectBox.js.
6  */
7  'use strict';
8  {
9      window SelectFilter = {
10         init: function(field_id, field_name, is_stacked) {
11             if (field_id match(/__prefix_/)) {
12                 // Don't initialize on empty forms.
13                 return;
14             }
15             const from_box = document.getElementById(field_id);
16             from_box.id += '_from'; // change its ID
17             from_box.className = 'filtered';
18
19             for (const p of from_box.parentNode.getElementsByTagName('p')) {
20                 if (p.classList.contains("info")) {
21                     // Remove <p class="info">, because it just gets in the way.
22                     from_box.parentNode.removeChild(p);
23                 } else if (p.classList.contains("help")) {
24                     // Move help text up to the top so it isn't below the select
25                     // boxes or wrapped off on the side to the right of the add
26                     // button:
27                     from_box.parentNode.insertBefore(p, from_box.parentNode.firstChild);
28                 }
29             }
30
31             // <div class="selector"> or <div class="selector stacked">
32             const selector_div = quickElement('div', from_box.parentNode);
33             selector_div.className = is_stacked ? 'selector stacked' : 'selector';
34
35             // <div class="selector-available">
36             const selector_available = quickElement('div', selector_div);
37             selector_available.className = 'selector-available';
38             const title_available = quickElement('h2', selector_available, interpolate(gettext('Available %s') + ' ', [field_name]));
39             quickElement(
40                 'span', title_available, " ",
41                 'class', 'help help-tooltip help-icon',
42                 'title', interpolate(
43                     gettext(
44                         'This is the list of available %s. You may choose some by ' +
45                         'selecting them in the box below and then clicking the ' +
46                         '"Choose" arrow between the two boxes.'
47                     ),
48                     [field_name]
49                 )
50             );
51
52             const filter_p = quickElement('p', selector_available, "", 'id', field_id + '_filter');
53             filter_p.className = 'selector-filter';
54
55             const search_filter_label = quickElement('label', filter_p, "", 'for', field_id + '_input');
56
57             quickElement(
58                 'span', search_filter_label, " ",
59                 'class', 'help-tooltip search-label-icon',
60                 'title', interpolate(gettext("Type into this box to filter down the list of available %s."), [field_name])
61             );
62
63             filter_p.appendChild(document.createTextNode(''));
64
65             const filter_input = quickElement('input', filter_p, "", 'type', 'text', 'placeholder', gettext("Filter"));
66             filter_input.id = field_id + '_input';
67
68             selector_available.appendChild(from_box);
69             const choose_all = quickElement('a', selector_available, gettext('Choose all'), 'title', interpolate(gettext('Click to choose all %s at once.'), [field_name]), 'href', '#', 'id', field_id + '_add_all');
70             choose_all.className = 'selector-chooseall';
71
72             // <ul class="selector-chooser">
73             const selector_chooser = quickElement('ul', selector_div);
74             selector_chooser.className = 'selector-chooser';
75             const add_link = quickElement('a', quickElement('li', selector_chooser), gettext('Choose'), 'title', gettext('Choose'), 'href', '#', 'id', field_id + '_add_link');
76             add_link.className = 'selector-add';
77             const remove_link = quickElement('a', quickElement('li', selector_chooser), gettext('Remove'), 'title', gettext('Remove'), 'href', '#', 'id', field_id + '_remove_link');
78             remove_link.className = 'selector-remove';
79
80             // <div class="selector-chosen">
81             const selector_chosen = quickElement('div', selector_div);
82             selector_chosen.className = 'selector-chosen';
83             const title_chosen = quickElement('h2', selector_chosen, interpolate(gettext('Chosen %s') + ' ', [field_name]));
84             quickElement(
85                 'span', title_chosen, " ",
86                 'class', 'help help-tooltip help-icon',
87                 'title', interpolate(
88                     gettext(
89                         'This is the list of chosen %s. You may remove some by ' +
90                         'selecting them in the box below and then clicking the ' +
91                         '"Remove" arrow between the two boxes.'
92                     ),
93                     [field_name]
94                 )
95             );
96
97             const to_box = quickElement('select', selector_chosen, "", 'id', field_id + '_to', 'multiple', "", 'size', from_box.size, 'name', from_box.name);
98             to_box.className = 'filtered';
99             const clear_all = quickElement('a', selector_chosen, gettext('Remove all'), 'title', interpolate(gettext('Click to remove all chosen %s at once.'), [field_name]), 'href', '#', 'id', field_id + '_re');
100             clear_all.className = 'selector-clearall';
101
102             from_box.name = from_box.name + '_old';
103
104             // Set up the JavaScript event handlers for the select box filter interface
105             const move_selection = function(e, elem, move_func, from, to) {
106                 if (elem.classList.contains('active')) {
107                     move_func(from, to);
108                     SelectFilter.refresh_icons(field_id);
109                 }
110                 e.preventDefault();
111             };
112             choose_all.addEventListener('click', function(e) {
113                 move_selection(e, this, SelectBox.move_all, field_id + '_from', field_id + '_to');
114             });
115             add_link.addEventListener('click', function(e) {

```

```

116     move_selection(e, this, SelectBox, move, field_id + '_from', field_id + '_to');
117 });
118 remove_link.addEventListener('click', function(e) {
119     move_selection(e, this, SelectBox, move, field_id + '_to', field_id + '_from');
120 });
121 clear_all.addEventListener('click', function(e) {
122     move_selection(e, this, SelectBox, move_all, field_id + '_to', field_id + '_from');
123 });
124 filter_input.addEventListener('keypress', function(e) {
125     SelectFilter.filter_key_press(e, field_id);
126 });
127 filter_input.addEventListener('keyup', function(e) {
128     SelectFilter.filter_key_up(e, field_id);
129 });
130 filter_input.addEventListener('keydown', function(e) {
131     SelectFilter.filter_key_down(e, field_id);
132 });
133 selector_div.addEventListener('change', function(e) {
134     if (e.target.tagName === 'SELECT') {
135         SelectFilter.refresh_icons(field_id);
136     }
137 });
138 selector_div.addEventListener('dblclick', function(e) {
139     if (e.target.tagName === 'OPTION') {
140         if (e.target.closest('select').id === field_id + '_to') {
141             SelectBox.move(field_id + '_to', field_id + '_from');
142         } else {
143             SelectBox.move(field_id + '_from', field_id + '_to');
144         }
145         SelectFilter.refresh_icons(field_id);
146     }
147 });
148 from_box.closest('form').addEventListener('submit', function() {
149     SelectBox.select_all(field_id + '_to');
150 });
151 SelectBox.init(field_id + '_from');
152 SelectBox.init(field_id + '_to');
153 // Move selected from_box options to to_box
154 SelectBox.move(field_id + '_from', field_id + '_to');
155
156 if (!is_stacked) {
157     // In horizontal mode, give the same height to the two boxes.
158     const j_from_box = document.getElementById(field_id + '_from');
159     const j_to_box = document.getElementById(field_id + '_to');
160     let height = filter_p.offsetHeight + j_from_box.offsetHeight;
161
162     const j_to_box_style = window.getComputedStyle(j_to_box);
163     if (j_to_box_style.getPropertyValue('box-sizing') === 'border-box') {
164         // Add the padding and border to the final height.
165         height += parseInt(j_to_box_style.getPropertyValue('padding-top'), 10)
166             + parseInt(j_to_box_style.getPropertyValue('padding-bottom'), 10)
167             + parseInt(j_to_box_style.getPropertyValue('border-top-width'), 10)
168             + parseInt(j_to_box_style.getPropertyValue('border-bottom-width'), 10);
169     }
170
171     j_to_box.style.height = height + 'px';
172 }
173
174 // Initial icon refresh
175 SelectFilter.refresh_icons(field_id);
176 },
177 any_selected: function(field) {
178     // Temporarily add the required attribute and check validity.
179     field.required = true;
180     const any_selected = field.checkValidity();
181     field.required = false;
182     return any_selected;
183 },
184 refresh_icons: function(field_id) {
185     const from = document.getElementById(field_id + '_from');
186     const to = document.getElementById(field_id + '_to');
187     // Active if at least one item is selected
188     document.getElementById(field_id + '_add_link').classList.toggle('active', SelectFilter.any_selected(from));
189     document.getElementById(field_id + '_remove_link').classList.toggle('active', SelectFilter.any_selected(to));
190     // Active if the corresponding box isn't empty
191     document.getElementById(field_id + '_add_all_link').classList.toggle('active', from.querySelector('option'));
192     document.getElementById(field_id + '_remove_all_link').classList.toggle('active', to.querySelector('option'));
193 },
194 filter_key_press: function(event, field_id) {
195     const from = document.getElementById(field_id + '_from');
196     // don't submit form if user pressed Enter
197     if ((event.which && event.which === 13) || (event.keyCode && event.keyCode === 13)) {
198         from.selectedIndex = 0;
199         SelectBox.move(field_id + '_from', field_id + '_to');
200         from.selectedIndex = 0;
201         event.preventDefault();
202     }
203 },
204 filter_key_up: function(event, field_id) {
205     const from = document.getElementById(field_id + '_from');
206     const temp = from.selectedIndex;
207     SelectBox.filter(field_id + '_from', document.getElementById(field_id + '_input').value);
208     from.selectedIndex = temp;
209 },
210 filter_key_down: function(event, field_id) {
211     const from = document.getElementById(field_id + '_from');
212     // right arrow -- move across
213     if ((event.which && event.which === 39) || (event.keyCode && event.keyCode === 39)) {
214         const old_index = from.selectedIndex;
215         SelectBox.move(field_id + '_from', field_id + '_to');
216         from.selectedIndex = (old_index === from.length) ? from.length - 1 : old_index;
217         return;
218     }
219     // down arrow -- wrap around
220     if ((event.which && event.which === 40) || (event.keyCode && event.keyCode === 40)) {
221         from.selectedIndex = (from.length === from.selectedIndex + 1) ? 0 : from.selectedIndex + 1;
222     }
223     // up arrow -- wrap around
224     if ((event.which && event.which === 38) || (event.keyCode && event.keyCode === 38)) {
225         from.selectedIndex = (from.selectedIndex === 0) ? from.length - 1 : from.selectedIndex - 1;
226     }
227 }
228 });
229
230 window.addEventListener('load', function(e) {

```

```

231     document.querySelectorAll('select.selectfilter, select.selectfilterstacked').forEach(function(el) {
232         const data = el.dataset;
233         SelectFilter.init(el.id, data.fieldName, parseInt(data.isStacked, 10));
234     });
235 });
236 }

```

#### File: ./static/admin/js/cancel.js

```

1  'use strict';
2  {
3      // Call function fn when the DOM is loaded and ready. If it is already
4      // loaded, call the function now.
5      // http://youmightnotneedjquery.com/#ready
6      function ready(fn) {
7          if (document.readyState !== 'loading') {
8              fn();
9          } else {
10             document.addEventListener('DOMContentLoaded', fn);
11         }
12     }
13
14     ready(function() {
15         function handleClick(event) {
16             event.preventDefault();
17             const params = new URLSearchParams(window.location.search);
18             if (params.has('popup')) {
19                 window.close(); // Close the popup.
20             } else {
21                 window.history.back(); // Otherwise, go back.
22             }
23         }
24
25         document.querySelectorAll('.cancel-link').forEach(function(el) {
26             el.addEventListener('click', handleClick);
27         });
28     });
29 }

```

#### File: ./static/admin/js/actions.js

```

1  /*global gettext, interpolate, ngettext*/
2  'use strict';
3  {
4      function show(selector) {
5          document.querySelectorAll(selector).forEach(function(el) {
6              el.classList.remove('hidden');
7          });
8      }
9
10     function hide(selector) {
11         document.querySelectorAll(selector).forEach(function(el) {
12             el.classList.add('hidden');
13         });
14     }
15
16     function showQuestion(options) {
17         hide(options.acrossClears);
18         show(options.acrossQuestions);
19         hide(options.allContainer);
20     }
21
22     function showClear(options) {
23         show(options.acrossClears);
24         hide(options.acrossQuestions);
25         document.querySelector(options.actionContainer).classList.remove(options.selectedClass);
26         show(options.allContainer);
27         hide(options.counterContainer);
28     }
29
30     function reset(options) {
31         hide(options.acrossClears);
32         hide(options.acrossQuestions);
33         hide(options.allContainer);
34         show(options.counterContainer);
35     }
36
37     function clearAcross(options) {
38         reset(options);
39         const acrossInputs = document.querySelectorAll(options.acrossInput);
40         acrossInputs.forEach(function(acrossInput) {
41             acrossInput.value = 0;
42         });
43         document.querySelector(options.actionContainer).classList.remove(options.selectedClass);
44     }
45
46     function checker(actionCheckboxes, options, checked) {
47         if (checked) {
48             showQuestion(options);
49         } else {
50             reset(options);
51         }
52         actionCheckboxes.forEach(function(el) {
53             el.checked = checked;
54             el.closest('tr').classList.toggle(options.selectedClass, checked);
55         });
56     }
57
58     function updateCounter(actionCheckboxes, options) {
59         const sel = Array.from(actionCheckboxes).filter(function(el) {
60             return el.checked;
61         }).length;
62         const counter = document.querySelector(options.counterContainer);
63         // data-actions-icnt is defined in the generated HTML
64         // and contains the total amount of objects in the queryset
65         const actions_icnt = Number(counter.dataset.actionsIcnt);
66         counter.textContent = interpolate(
67             ngettext('%(sel)s of %(cnt)s selected', '%(sel)s of %(cnt)s selected', sel), {
68                 sel: sel,

```

```

69     cnt: actions_jcnt
70   }, true);
71   const allToggle = document.getElementById('options allToggleId');
72   allToggle.checked = sel === actionCheckboxes.length;
73   if (allToggle.checked) {
74     showQuestion(options);
75   } else {
76     clearAcross(options);
77   }
78 }
79
80 const defaults = {
81   actionContainer: "div.actions",
82   counterContainer: "span.action-counter",
83   allContainer: "div.actions span.all",
84   acrossInput: "div.actions input.select-across",
85   acrossQuestions: "div.actions span.question",
86   acrossClears: "div.actions span.clear",
87   allToggleId: "action-toggle",
88   selectedClass: "selected"
89 };
90
91 window.Actions = function(actionCheckboxes, options) {
92   options = Object.assign({}, defaults, options);
93   let list_editable_changed = false;
94   let lastChecked = null;
95   let shiftPressed = false;
96
97   document.addEventListener('keydown', (event) => {
98     shiftPressed = event.shiftKey;
99   });
100
101   document.addEventListener('keyup', (event) => {
102     shiftPressed = event.shiftKey;
103   });
104
105   document.getElementById('options allToggleId').addEventListener('click', function(event) {
106     checker.actionCheckboxes = options, this.checked;
107     updateCounter(actionCheckboxes, options);
108   });
109
110   document.querySelectorAll('options acrossQuestions + " a"').forEach(function(el) {
111     el.addEventListener('click', function(event) {
112       event.preventDefault();
113       const acrossInputs = document.querySelectorAll('options acrossInput');
114       acrossInputs.forEach(function(acrossInput) {
115         acrossInput.value = 1;
116       });
117       showClear(options);
118     });
119   });
120
121   document.querySelectorAll('options acrossClears + " a"').forEach(function(el) {
122     el.addEventListener('click', function(event) {
123       event.preventDefault();
124       document.getElementById('options allToggleId').checked = false;
125       clearAcross(options);
126       checker.actionCheckboxes = options, false;
127       updateCounter(actionCheckboxes, options);
128     });
129   });
130
131   function affectedCheckboxes(target, withModifier) {
132     const multiSelect = !lastChecked && withModifier && lastChecked !== target;
133     if (!multiSelect) {
134       return [target];
135     }
136     const checkboxes = Array.from(actionCheckboxes);
137     const targetIndex = checkboxes.findIndex(el => el === target);
138     const lastCheckedIndex = checkboxes.findIndex(el => el === lastChecked);
139     const startIndex = Math.min(targetIndex, lastCheckedIndex);
140     const endIndex = Math.max(targetIndex, lastCheckedIndex);
141     const filtered = checkboxes.filter((el, index) => (startIndex <= index) && (index <= endIndex));
142     return filtered;
143   }
144
145   Array.from(document.getElementById('result_list').tbody).forEach(function(el) {
146     el.addEventListener('change', function(event) {
147       const target = event.target;
148       if (target.classList.contains('action-select')) {
149         const checkboxes = affectedCheckboxes(target, shiftPressed);
150         checker.checkboxes = options, target, checked;
151         updateCounter(actionCheckboxes, options);
152         lastChecked = target;
153       } else {
154         list_editable_changed = true;
155       }
156     });
157   });
158
159   document.querySelector('#changelist-form button[name=index]').addEventListener('click', function(event) {
160     if (list_editable_changed) {
161       const confirmed = confirm(gettext("You have unsaved changes on individual editable fields. If you run an action, your unsaved changes will be lost."));
162       if (!confirmed) {
163         event.preventDefault();
164       }
165     }
166   });
167
168   const el = document.querySelector('#changelist-form input[name=_save]');
169   // The button does not exist if no fields are editable.
170   if (el) {
171     el.addEventListener('click', function(event) {
172       if (document.querySelector("[name=action]").value) {
173         const text = list_editable_changed
174           ? gettext("You have selected an action, but you havent saved your changes to individual fields yet. Please click OK to save. Youll need to re-run the action.")
175           : gettext("You have selected an action, and you havent made any changes on individual fields. Youre probably looking for the Go button rather than the Save button.");
176         if (!confirm(text)) {
177           event.preventDefault();
178         }
179       }
180     });
181   }
182 };
183

```

```

184 // Call function fn when the DOM is loaded and ready. If it is already
185 // loaded, call the function now.
186 // http://youmightnotneedjquery.com/#ready
187 function ready(fn) {
188     if (document.readyState !== 'loading') {
189         fn();
190     } else {
191         document.addEventListener('DOMContentLoaded', fn);
192     }
193 }
194
195 ready(function() {
196     const actionsEls = document.querySelectorAll('tr input.action-select');
197     if (actionsEls.length > 0) {
198         Actions(actionsEls);
199     }
200 });
201 }

```

**File: ./static/admin/js/admin/DateTimeShortcuts.js**

```

1  /*global Calendar, findPosX, findPosY, get_format, gettext, gettext_noop, interpolate, ngettext, quickElement*/
2  // Inserts shortcut buttons after all of the following:
3  // <input type="text" class="vDateField">
4  // <input type="text" class="vTimeField">
5  'use strict';
6  {
7      const DateTimeShortcuts = {
8          calendars: [],
9          calendarInputs: [],
10         clockInputs: [],
11         clockHours: {
12             default: [
13                 [gettext_noop('Now'), -1],
14                 [gettext_noop('Midnight'), 0],
15                 [gettext_noop('6 a.m.'), 6],
16                 [gettext_noop('Noon'), 12],
17                 [gettext_noop('6 p.m.'), 18]
18             ]
19         },
20         dismissClockFunc: [],
21         dismissCalendarFunc: [],
22         calendarDivName1: 'calendarbox', // name of calendar <div> that gets toggled
23         calendarDivName2: 'calendarin', // name of <div> that contains calendar
24         calendarLinkName: 'calendarlink', // name of the link that is used to toggle
25         clockDivName: 'clockbox', // name of clock <div> that gets toggled
26         clockLinkName: 'clocklink', // name of the link that is used to toggle
27         shortCutsClass: 'datetimeshortcuts', // class of the clock and cal shortcuts
28         timeZoneWarningClass: 'timezonewarning', // class of the warning for timezone mismatch
29         timeZoneOffset: 0,
30         init: function() {
31             const serverOffset = document.body.dataset.adminUtcOffset;
32             if (serverOffset) {
33                 const localOffset = new Date().getTimezoneOffset() * -60;
34                 DateTimeShortcuts.timeZoneOffset = localOffset - serverOffset;
35             }
36
37             for (const inp of document.getElementsByTagName('input')) {
38                 if (inp.type === 'text' && inp.classList.contains('vTimeField')) {
39                     DateTimeShortcuts.addClock(inp);
40                     DateTimeShortcuts.addTimezoneWarning(inp);
41                 }
42                 else if (inp.type === 'text' && inp.classList.contains('vDateField')) {
43                     DateTimeShortcuts.addCalendar(inp);
44                     DateTimeShortcuts.addTimezoneWarning(inp);
45                 }
46             }
47         },
48         // Return the current time while accounting for the server timezone.
49         now: function() {
50             const serverOffset = document.body.dataset.adminUtcOffset;
51             if (serverOffset) {
52                 const localNow = new Date();
53                 const localOffset = localNow.getTimezoneOffset() * -60;
54                 localNow.setTime(localNow.getTime() + 1000 * (serverOffset - localOffset));
55                 return localNow;
56             }
57             else {
58                 return new Date();
59             }
60         },
61         // Add a warning when the time zone in the browser and backend do not match.
62         addTimezoneWarning: function(inp) {
63             const warningClass = DateTimeShortcuts.timeZoneWarningClass;
64             let timeZoneOffset = DateTimeShortcuts.timeZoneOffset / 3600;
65
66             // Only warn if there is a time zone mismatch.
67             if (!timeZoneOffset) {
68                 return;
69             }
70
71             // Check if warning is already there.
72             if (inp.parentNode.querySelectorAll('.' + warningClass).length) {
73                 return;
74             }
75
76             let message;
77             if (timeZoneOffset > 0) {
78                 message = ngettext(
79                     'Note: You are %s hour ahead of server time.',
80                     'Note: You are %s hours ahead of server time.',
81                     timeZoneOffset
82                 );
83             }
84             else {
85                 timeZoneOffset *= -1;
86                 message = ngettext(
87                     'Note: You are %s hour behind server time.',
88                     'Note: You are %s hours behind server time.',
89                     timeZoneOffset
90                 );
91             }
92         }
93     };
94 }

```

```

90     }
91     message = interpolate message, {timezoneOffset});
92
93     const warning = document.createElement('span');
94     warning.className = warningClass;
95     warning.textContent = message;
96     inp.parentNode.appendChild(document.createElement('br'));
97     inp.parentNode.appendChild(warning);
98 }
99 // Add clock widget to a given field
100 addClock: function(inp) {
101     const num = DateTimeShortcuts.clockInputs.length;
102     DateTimeShortcuts.clockInputs[num] = inp;
103     DateTimeShortcuts.dismissClockFunc[num] = function() { DateTimeShortcuts.dismissClock(num); return true; };
104
105     // Shortcut links (clock icon and "Now" link)
106     const shortcuts_span = document.createElement('span');
107     shortcuts_span.className = DateTimeShortcuts.shortCutsClass;
108     inp.parentNode.insertBefore(shortcuts_span, inp.nextSibling);
109     const now_link = document.createElement('a');
110     now_link.href = "#";
111     now_link.textContent = gettext('Now');
112     now_link.addEventListener('click', function(e) {
113         e.preventDefault();
114         DateTimeShortcuts.handleClickQuicklink(num, -1);
115     });
116     const clock_link = document.createElement('a');
117     clock_link.href = "#";
118     clock_link.id = DateTimeShortcuts.clockLinkName + num;
119     clock_link.addEventListener('click', function(e) {
120         e.preventDefault();
121         // avoid triggering the document click handler to dismiss the clock
122         e.stopPropagation();
123         DateTimeShortcuts.openClock(num);
124     });
125
126     quickElement(
127         'span', clock_link, "
128         'class', 'clock-icon',
129         'title', gettext('Choose a Time')
130     );
131     shortcuts_span.appendChild(document.createTextNode("\u00A0"));
132     shortcuts_span.appendChild(now_link);
133     shortcuts_span.appendChild(document.createTextNode("\u00A0\u00A0"));
134     shortcuts_span.appendChild(clock_link);
135
136     // Create clock link div
137     //
138     // Markup looks like:
139     // <div id="clockbox1" class="clockbox module">
140     //   <h2>Choose a time</h2>
141     //   <ul class="timelist">
142     //     <li><a href="#">Now</a></li>
143     //     <li><a href="#">Midnight</a></li>
144     //     <li><a href="#">6 a.m.</a></li>
145     //     <li><a href="#">Noon</a></li>
146     //     <li><a href="#">6 p.m.</a></li>
147     //   </ul>
148     //   <p class="calendar-cancel"><a href="#">Cancel</a></p>
149     // </div>
150
151     const clock_box = document.createElement('div');
152     clock_box.style.display = 'none';
153     clock_box.style.position = 'absolute';
154     clock_box.className = 'clockbox module';
155     clock_box.id = DateTimeShortcuts.clockDivName + num;
156     document.body.appendChild(clock_box);
157     clock_box.addEventListener('click', function(e) { e.stopPropagation(); });
158
159     quickElement('h2', clock_box, gettext('Choose a time'));
160     const time_list = quickElement('ul', clock_box);
161     time_list.className = 'timelist';
162     // The list of choices can be overridden in JavaScript like this:
163     // DateTimeShortcuts.clockHours.name = ['3 a.m.', 3];
164     // where name is the name attribute of the <input>.
165     const name = typeof DateTimeShortcuts.clockHours[inp.name] === 'undefined' ? 'default_' : inp.name;
166     DateTimeShortcuts.clockHours[name].forEach(function(element) {
167         const time_link = quickElement('a', quickElement('li', time_list), gettext(element[0]), 'href', '#');
168         time_link.addEventListener('click', function(e) {
169             e.preventDefault();
170             DateTimeShortcuts.handleClickQuicklink(num, element[1]);
171         });
172     });
173
174     const cancel_p = quickElement('p', clock_box);
175     cancel_p.className = 'calendar-cancel';
176     const cancel_link = quickElement('a', cancel_p, gettext('Cancel'), 'href', '#');
177     cancel_link.addEventListener('click', function(e) {
178         e.preventDefault();
179         DateTimeShortcuts.dismissClock(num);
180     });
181
182     document.addEventListener('keyup', function(event) {
183         if (event.which === 27) {
184             // ESC key closes popup
185             DateTimeShortcuts.dismissClock(num);
186             event.preventDefault();
187         }
188     });
189 },
190 openClock: function(num) {
191     const clock_box = document.getElementById(DateTimeShortcuts.clockDivName + num);
192     const clock_link = document.getElementById(DateTimeShortcuts.clockLinkName + num);
193
194     // Recalculate the clockbox position
195     // is it left-to-right or right-to-left layout ?
196     if (window.getComputedStyle(document.body).direction !== 'rtl') {
197         clock_box.style.left = findPosX(clock_link) + 17 + 'px';
198     }
199     else {
200         // since style's width is in em, it'd be tough to calculate
201         // px value of it. let's use an estimated px for now
202         clock_box.style.left = findPosX(clock_link) - 110 + 'px';
203     }
204     clock_box.style.top = Math.max(0, findPosY(clock_link) - 30) + 'px';

```

```

205
206 // Show the clock box
207 clock_box.style.display = 'block';
208 document.addEventListener('click', DateTimeShortcuts.dismissClockFunc(num));
209 },
210 dismissClock: function(num) {
211     document.getElementById(DateTimeShortcuts.clockDivName + num).style.display = 'none';
212     document.removeEventListener('click', DateTimeShortcuts.dismissClockFunc(num));
213 },
214 handleClockQuicklink: function(num, val) {
215     let d;
216     if (val === -1) {
217         d = DateTimeShortcuts.now();
218     }
219     else {
220         d = new Date(1970, 1, 1, val, 0, 0, 0);
221     }
222     DateTimeShortcuts.clockInputs[num].value = d.strftime.get_format('TIME_INPUT_FORMATS')[0];
223     DateTimeShortcuts.clockInputs[num].focus();
224     DateTimeShortcuts.dismissClock(num);
225 },
226 // Add calendar widget to a given field.
227 addCalendar: function(inp) {
228     const num = DateTimeShortcuts.calendars.length;
229
230     DateTimeShortcuts.calendarInputs[num] = inp;
231     DateTimeShortcuts.dismissCalendarFunc[num] = function() { DateTimeShortcuts.dismissCalendar(num); return true; };
232
233     // Shortcut links (calendar icon and "Today" link)
234     const shortcuts_span = document.createElement('span');
235     shortcuts_span.className = DateTimeShortcuts.shortCutsClass;
236     inp.parentNode.insertBefore(shortcuts_span, inp.nextSibling);
237     const today_link = document.createElement('a');
238     today_link.href = '#';
239     today_link.appendChild(document.createTextNode(gettext('Today')));
240     today_link.addEventListener('click', function(e) {
241         e.preventDefault();
242         DateTimeShortcuts.handleCalendarQuickLink(num, 0);
243     });
244     const cal_link = document.createElement('a');
245     cal_link.href = '#';
246     cal_link.id = DateTimeShortcuts.calendarLinkName + num;
247     cal_link.addEventListener('click', function(e) {
248         e.preventDefault();
249         // avoid triggering the document click handler to dismiss the calendar
250         e.stopPropagation();
251         DateTimeShortcuts.openCalendar(num);
252     });
253     quickElement(
254         'span', cal_link, "
255         'class', 'date-icon',
256         'title', gettext('Choose a Date')
257     );
258     shortcuts_span.appendChild(document.createTextNode('\u00A0'));
259     shortcuts_span.appendChild(today_link);
260     shortcuts_span.appendChild(document.createTextNode('\u00A0\u00A0'));
261     shortcuts_span.appendChild(cal_link);
262
263     // Create calendarbox div.
264     //
265     // Markup looks like:
266     //
267     // <div id="calendarbox3" class="calendarbox module">
268     //   <h2>
269     //     <a href="#" class="link-previous">&lsaquo;</a></h2>
270     //     <a href="#" class="link-next">&rsaquo;</a> February 2003
271     //   </h2>
272     //   <div class="calendar" id="calendarin3">
273     //     <!-- (cal) -->
274     //   </div>
275     //   <div class="calendar-shortcuts">
276     //     <a href="#">Yesterday</a> | <a href="#">Today</a> | <a href="#">Tomorrow</a>
277     //   </div>
278     //   <p class="calendar-cancel"><a href="#">Cancel</a></p>
279     // </div>
280     const cal_box = document.createElement('div');
281     cal_box.style.display = 'none';
282     cal_box.style.position = 'absolute';
283     cal_box.className = 'calendarbox module';
284     cal_box.id = DateTimeShortcuts.calendarDivName1 + num;
285     document.body.appendChild(cal_box);
286     cal_box.addEventListener('click', function(e) { e.stopPropagation(); });
287
288     // next-prev links
289     const cal_nav = quickElement('div', cal_box);
290     const cal_nav_prev = quickElement('a', cal_nav, '<', 'href', '#');
291     cal_nav_prev.className = 'calendarnav-previous';
292     cal_nav_prev.addEventListener('click', function(e) {
293         e.preventDefault();
294         DateTimeShortcuts.drawPrev(num);
295     });
296
297     const cal_nav_next = quickElement('a', cal_nav, '>', 'href', '#');
298     cal_nav_next.className = 'calendarnav-next';
299     cal_nav_next.addEventListener('click', function(e) {
300         e.preventDefault();
301         DateTimeShortcuts.drawNext(num);
302     });
303
304     // main box
305     const cal_main = quickElement('div', cal_box, 'id', DateTimeShortcuts.calendarDivName2 + num);
306     cal_main.className = 'calendar';
307     DateTimeShortcuts.calendars[num] = new Calendar(DateTimeShortcuts.calendarDivName2 + num, DateTimeShortcuts.handleCalendarCallback(num));
308     DateTimeShortcuts.calendars[num].drawCurrent();
309
310     // calendar shortcuts
311     const shortcuts = quickElement('div', cal_box);
312     shortcuts.className = 'calendar-shortcuts';
313     let day_link = quickElement('a', shortcuts, gettext('Yesterday'), 'href', '#');
314     day_link.addEventListener('click', function(e) {
315         e.preventDefault();
316         DateTimeShortcuts.handleCalendarQuickLink(num, -1);
317     });
318     shortcuts.appendChild(document.createTextNode('\u00A0\u00A0'));
319     day_link = quickElement('a', shortcuts, gettext('Today'), 'href', '#');

```

```

320 day_link.addEventListener('click', function(e) {
321     e.preventDefault();
322     DateTimeShortcuts.handleCalendarQuickLink(num, 0);
323 });
324 shortcuts.appendChild(document.createTextNode("\u00A0\u00A0"));
325 day_link = quickElement('a', shortcuts, 'Tomorrow', 'href', '#');
326 day_link.addEventListener('click', function(e) {
327     e.preventDefault();
328     DateTimeShortcuts.handleCalendarQuickLink(num, +1);
329 });
330
331 // cancel bar
332 const cancel_p = quickElement('p', cal_box);
333 cancel_p.className = 'calendar-cancel';
334 const cancel_link = quickElement('a', cancel_p, 'Cancel', 'href', '#');
335 cancel_link.addEventListener('click', function(e) {
336     e.preventDefault();
337     DateTimeShortcuts.dismissCalendar(num);
338 });
339 document.addEventListener('keyup', function(event) {
340     if (event.which === 27) {
341         // ESC key closes popup
342         DateTimeShortcuts.dismissCalendar(num);
343         event.preventDefault();
344     }
345 });
346 },
347 openCalendar: function(num) {
348     const cal_box = document.getElementById(DateTimeShortcuts.calendarDivName1 + num);
349     const cal_link = document.getElementById(DateTimeShortcuts.calendarLinkName + num);
350     const inp = DateTimeShortcuts.calendarInputs[num];
351
352     // Determine if the current value in the input has a valid date.
353     // If so, draw the calendar with that date's year and month.
354     if (inp.value) {
355         const format = get_format('DATE_INPUT_FORMATS')[0];
356         const selected = inp.value.strptime(format);
357         const year = selected.getUTCFullYear();
358         const month = selected.getUTCMonth() + 1;
359         const re = /\d{4}/;
360         if (re.test(year.toString()) && month >= 1 && month <= 12) {
361             DateTimeShortcuts.calendars[num].drawDate(month, year, selected);
362         }
363     }
364
365     // Recalculate the clockbox position
366     // is it left-to-right or right-to-left layout ?
367     if (window.getComputedStyle(document.body).direction !== 'rtl') {
368         cal_box.style.left = findPosX(cal_link) + 17 + 'px';
369     }
370     else {
371         // since style's width is in em, it'd be tough to calculate
372         // px value of it. let's use an estimated px for now
373         cal_box.style.left = findPosX(cal_link) - 180 + 'px';
374     }
375     cal_box.style.top = Math.max(0, findPosY(cal_link) - 75) + 'px';
376
377     cal_box.style.display = 'block';
378     document.addEventListener('click', DateTimeShortcuts.dismissCalendarFunc(num));
379 },
380 dismissCalendar: function(num) {
381     document.getElementById(DateTimeShortcuts.calendarDivName1 + num).style.display = 'none';
382     document.removeEventListener('click', DateTimeShortcuts.dismissCalendarFunc(num));
383 },
384 drawPrev: function(num) {
385     DateTimeShortcuts.calendars[num].drawPreviousMonth();
386 },
387 drawNext: function(num) {
388     DateTimeShortcuts.calendars[num].drawNextMonth();
389 },
390 handleCalendarCallback: function(num) {
391     let format = get_format('DATE_INPUT_FORMATS')[0];
392     // the format needs to be escaped a little
393     format = format.replace('\\', '\\\\')
394         .replace('\r', '\\r')
395         .replace('\n', '\\n')
396         .replace('\t', '\\t')
397         .replace('"', '\\"');
398     return function(y, m, d) {
399         DateTimeShortcuts.calendarInputs[num].value = new Date(y, m - 1, d).strftime(format);
400         DateTimeShortcuts.calendarInputs[num].focus();
401         document.getElementById(DateTimeShortcuts.calendarDivName1 + num).style.display = 'none';
402     };
403 },
404 handleCalendarQuickLink: function(num, offset) {
405     const d = DateTimeShortcuts.now();
406     d.setDate(d.getDate() + offset);
407     DateTimeShortcuts.calendarInputs[num].value = d.strftime(get_format('DATE_INPUT_FORMATS')[0]);
408     DateTimeShortcuts.calendarInputs[num].focus();
409     DateTimeShortcuts.dismissCalendar(num);
410 }
411 };
412
413 window.addEventListener('load', DateTimeShortcuts.init);
414 window.DateTimeShortcuts = DateTimeShortcuts;
415 }

```

#### File: ./static/admin/js/admin/RelatedObjectLookups.js

```

1  /*global SelectBox, interpolate*/
2  // Handles related-objects functionality: lookup link for raw_id_fields
3  // and Add Another links.
4  'use strict';
5  {
6      const $ = django.jQuery;
7
8      function showAdminPopup(triggeringLink, name_regexp, add_popup) {
9          const name = triggeringLink.id.replace(name_regexp, '');

```



```

10     const href = new URL(triggeringLink.href);
11     if (add_popup) {
12         href.searchParams.set('_popup', 1);
13     }
14     const win = window.open(href, name, 'height=500,width=800,resizable=yes,scrollbars=yes');
15     win.focus();
16     return false;
17 }
18
19 function showRelatedObjectLookupPopup(triggeringLink) {
20     return showAdminPopup(triggeringLink, /^lookup$/, true);
21 }
22
23 function dismissRelatedLookupPopup(win, chosenId) {
24     const name = win.name;
25     const elem = document.getElementById(name);
26     if (elem.classList.contains('vManyToManyRawIdAdminField') && elem.value) {
27         elem.value += ',' + chosenId;
28     } else {
29         document.getElementById(name).value = chosenId;
30     }
31     win.close();
32 }
33
34 function showRelatedObjectPopup(triggeringLink) {
35     return showAdminPopup(triggeringLink, /^(change|add|delete)$/, false);
36 }
37
38 function updateRelatedObjectLinks(triggeringLink) {
39     const $this = $(triggeringLink);
40     const siblings = $this.nextAll('.view-related, .change-related, .delete-related');
41     if (!siblings.length) {
42         return;
43     }
44     const value = $this.val();
45     if (value) {
46         siblings.each(function() {
47             const elm = $(this);
48             elm.attr('href', elm.attr('data-href-template').replace('__fk__', value));
49         });
50     } else {
51         siblings.removeAttr('href');
52     }
53 }
54
55 function dismissAddRelatedObjectPopup(win, newId, newRepr) {
56     const name = win.name;
57     const elem = document.getElementById(name);
58     if (elem) {
59         const elemName = elem.nodeName.toUpperCase();
60         if (elemName === 'SELECT') {
61             elem.options[elem.options.length] = new Option(newRepr, newId, true, true);
62         } else if (elemName === 'INPUT') {
63             if (elem.classList.contains('vManyToManyRawIdAdminField') && elem.value) {
64                 elem.value += ',' + newId;
65             } else {
66                 elem.value = newId;
67             }
68         }
69         // Trigger a change event to update related links if required.
70         $elem.trigger('change');
71     } else {
72         const told = name + "_to";
73         const o = new Option(newRepr, newId);
74         SelectBox.add_to_cache(told, o);
75         SelectBox.redisplay(told);
76     }
77     win.close();
78 }
79
80 function dismissChangeRelatedObjectPopup(win, objId, newRepr, newId) {
81     const id = win.name.replace(/^edit$/, '');
82     const selectsSelector = interpolate('%s, %s_from, %s_to', [id, id, id]);
83     const selects = $(selectsSelector);
84     selects.find('option').each(function() {
85         if (this.value === objId) {
86             this.textContent = newRepr;
87             this.value = newId;
88         }
89     });
90     selects.next().find('.select2-selection__rendered').each(function() {
91         // The element can have a clear button as a child.
92         // Use the lastChild to modify only the displayed value.
93         this.lastChild.textContent = newRepr;
94         this.title = newRepr;
95     });
96     win.close();
97 }
98
99 function dismissDeleteRelatedObjectPopup(win, objId) {
100     const id = win.name.replace(/^delete$/, '');
101     const selectsSelector = interpolate('%s, %s_from, %s_to', [id, id, id]);
102     const selects = $(selectsSelector);
103     selects.find('option').each(function() {
104         if (this.value === objId) {
105             $(this).remove();
106         }
107     });
108     win.close();
109 }
110
111 window.showRelatedObjectLookupPopup = showRelatedObjectLookupPopup;
112 window.dismissRelatedLookupPopup = dismissRelatedLookupPopup;
113 window.showRelatedObjectPopup = showRelatedObjectPopup;
114 window.updateRelatedObjectLinks = updateRelatedObjectLinks;
115 window.dismissAddRelatedObjectPopup = dismissAddRelatedObjectPopup;
116 window.dismissChangeRelatedObjectPopup = dismissChangeRelatedObjectPopup;
117 window.dismissDeleteRelatedObjectPopup = dismissDeleteRelatedObjectPopup;
118
119 // Kept for backward compatibility
120 window.showAddAnotherPopup = showRelatedObjectPopup;
121 window.dismissAddAnotherPopup = dismissAddRelatedObjectPopup;
122
123 $(document).ready(function() {
124     $('[data-popup-opener]') on('click', function(event) {

```

```

125     event.preventDefault();
126     opener.dismissRelatedLookupPopup(window, $(this).data("popup-opener"));
127 });
128 $('body').on('click', '.related-widget-wrapper-link', function(e) {
129     e.preventDefault();
130     if (this.href) {
131         const event = $.Event('django:show-related', {href: this.href});
132         $(this).trigger(event);
133         if (!event.isDefaultPrevented()) {
134             showRelatedObjectPopup(this);
135         }
136     }
137 });
138 $('body').on('change', '.related-widget-wrapper select', function(e) {
139     const event = $.Event('django:update-related');
140     $(this).trigger(event);
141     if (!event.isDefaultPrevented()) {
142         updateRelatedObjectLinks(this);
143     }
144 });
145 $('.related-widget-wrapper select').trigger('change');
146 $('body').on('click', '.related-lookup', function(e) {
147     e.preventDefault();
148     const event = $.Event('django:lookup-related');
149     $(this).trigger(event);
150     if (!event.isDefaultPrevented()) {
151         showRelatedObjectLookupPopup(this);
152     }
153 });
154 });
155 }

```

#### File: ./static/admin/js/prepopulate\_init.js

```

1  'use strict';
2  {
3      const $ = django.jQuery;
4      const fields = $('#django-admin-prepopulated-fields-constants').data('prepopulatedFields');
5      $each(fields, function(index, field) {
6          $('#empty-form .form-row .field-' + field.name + ', .empty-form .form-row .field-' + field.name).addClass('prepopulated_field');
7          $field[id].data('dependency_list', field.dependency_list).prepopulate(
8              field.dependency_ids, field.maxLength, field.allowUnicode
9          );
10     });
11 }

```

#### File: ./static/admin/js/urlify.js

```

1  /*global XRegExp*/
2  'use strict';
3  {
4      const LATIN_MAP = {
5          'À': 'A', 'Á': 'A', 'Â': 'A', 'Ã': 'A', 'Ä': 'A', 'Å': 'AE',
6          'Ç': 'C', 'È': 'E', 'É': 'E', 'Ê': 'E', 'Ë': 'E', 'Ï': 'I',
7          'Ð': 'D', 'Ñ': 'N', 'Ò': 'O', 'Ó': 'O', 'Ô': 'O',
8          'Õ': 'O', 'Ö': 'O', 'Ø': 'O', 'Ù': 'U', 'Ú': 'U', 'Û': 'U',
9          'Ü': 'U', 'Ý': 'Y', 'Þ': 'T', 'ß': 'ss', 'à': 'a',
10         'á': 'a', 'â': 'a', 'ã': 'a', 'ä': 'ae', 'å': 'a',
11         'ç': 'c', 'è': 'e', 'é': 'e', 'ê': 'e', 'ë': 'e', 'ï': 'i',
12         'ð': 'd', 'ñ': 'n', 'ó': 'o', 'ô': 'o', 'õ': 'o',
13         'ö': 'o', 'ø': 'o', 'ù': 'u', 'ú': 'u', 'û': 'u',
14         'ü': 'y', 'ý': 'y', 'þ': 't', 'ÿ': 'y'
15     };
16     const LATIN_SYMBOLS_MAP = {
17         '©': 'c'
18     };
19     const GREEK_MAP = {
20         'α': 'a', 'β': 'b', 'γ': 'g', 'δ': 'd', 'ε': 'e', 'ζ': 'z', 'η': 'h',
21         'θ': 't', 'ι': 'i', 'κ': 'k', 'λ': 'l', 'μ': 'm', 'ν': 'n', 'ξ': '3',
22         'ο': 'o', 'π': 'p', 'ρ': 'r', 'σ': 's', 'τ': 't', 'υ': 'y', 'φ': 'f',
23         'χ': 'x', 'ψ': 'ps', 'ω': 'w', 'αι': 'ae', 'oi': 'oi',
24         'γ': 'y', 'η': 'h', 'ω': 'w', 'ς': 's', 'φ': 'f', 'ψ': 'y',
25         'Α': 'A', 'Β': 'B', 'Γ': 'G', 'Δ': 'D', 'Ε': 'E', 'Ζ': 'Z',
26         'Η': 'H', 'Θ': 't', 'Ι': 'I', 'Κ': 'K', 'Λ': 'L', 'Μ': 'M', 'Ν': 'N',
27         'Ξ': '3', 'Ο': 'O', 'Π': 'P', 'Ρ': 'R', 'Σ': 'S', 'Τ': 'T', 'Υ': 'Y',
28         'Φ': 'F', 'Χ': 'X', 'Ψ': 'PS', 'Ω': 'W', 'Α': 'A', 'Ε': 'E', 'Ι': 'I',
29         'Ο': 'O', 'Υ': 'Y', 'Η': 'H', 'Ω': 'W', 'Π': 'P', 'Υ': 'Y'
30     };
31     const TURKISH_MAP = {
32         'ş': 's', 'Ş': 'S', 'ı': 'i', 'İ': 'I', 'ç': 'c', 'Ç': 'C', 'ü': 'u',
33         'Ü': 'U', 'ö': 'o', 'Ö': 'O', 'ğ': 'g', 'Ğ': 'G'
34     };
35     const ROMANIAN_MAP = {
36         'ă': 'a', 'â': 'a', 'î': 'i', 'ș': 's', 'ț': 't', 'â': 'a',
37         'Ă': 'A', 'Â': 'A', 'Î': 'I', 'Ș': 'S', 'Ț': 'T', 'Â': 'A'
38     };
39     const RUSSIAN_MAP = {
40         'а': 'a', 'б': 'b', 'в': 'v', 'г': 'g', 'д': 'd', 'е': 'e', 'ё': 'yo',
41         'з': 'z', 'и': 'i', 'й': 'j', 'к': 'k', 'л': 'l', 'м': 'm',
42         'н': 'n', 'о': 'o', 'п': 'p', 'р': 'r', 'с': 's', 'т': 't', 'у': 'u',
43         'ф': 'f', 'х': 'h', 'ц': 'c', 'ч': 'ch', 'ш': 'sh', 'щ': 'sh', 'ъ': 'x',
44         'ы': 'y', 'э': 'e', 'ю': 'yu', 'я': 'ya',
45         'А': 'A', 'Б': 'B', 'В': 'V', 'Г': 'G', 'Д': 'D', 'Е': 'E', 'Ё': 'Yo',
46         'З': 'Z', 'И': 'I', 'Й': 'J', 'К': 'K', 'Л': 'L', 'М': 'M',
47         'Н': 'N', 'О': 'O', 'П': 'P', 'Р': 'R', 'С': 'S', 'Т': 'T', 'У': 'U',
48         'Ф': 'F', 'Х': 'H', 'Ц': 'C', 'Ч': 'Ch', 'Ш': 'Sh', 'Щ': 'Sh', 'Ъ': 'X',
49         'Ы': 'Y', 'Э': 'E', 'Ю': 'Yu', 'Я': 'Ya'
50     };
51     const UKRAINIAN_MAP = {
52         'є': 'Ye', 'І': 'I', 'Ї': 'Yi', 'Г': 'G', 'є': 'Ye', 'ї': 'I',
53         'й': 'g'
54     };
55     const CZECH_MAP = {
56         'č': 'c', 'ď': 'd', 'ě': 'e', 'ň': 'n', 'ř': 'r', 'š': 's', 'ť': 't',
57         'ů': 'u', 'ž': 'z', 'Č': 'C', 'Ď': 'D', 'Ě': 'E', 'Ň': 'N', 'Ř': 'R',
58         'Š': 'S', 'Ť': 'T', 'Ů': 'U', 'Ž': 'Z'
59     };
60     const SLOVAK_MAP = {
61         'ä': 'a', 'ä': 'a', 'ä': 'c', 'ä': 'd', 'ä': 'e', 'ä': 'i', 'ä': 'l',

```

```

62     "ı", "İ", "ı", "n", "ı", "o", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
63     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
64     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
65     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
66     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
67 };
68 const POLISH_MAP = {
69     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
70     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
71     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
72     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
73 };
74 const LATVIAN_MAP = {
75     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
76     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
77     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
78     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
79 };
80 const ARABIC_MAP = {
81     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
82     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
83     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
84     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
85 };
86 const LITHUANIAN_MAP = {
87     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
88     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
89     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
90     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
91 };
92 const SERBIAN_MAP = {
93     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
94     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
95     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
96 };
97 const AZERBAIJANI_MAP = {
98     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
99     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
100 };
101 const GEORGIAN_MAP = {
102     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
103     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
104     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
105     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
106     "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı", "ı",
107 };
108
109 const ALL_DOWNCODE_MAPS = [
110     LATIN_MAP,
111     LATIN_SYMBOLS_MAP,
112     GREEK_MAP,
113     TURKISH_MAP,
114     ROMANIAN_MAP,
115     RUSSIAN_MAP,
116     UKRAINIAN_MAP,
117     CZECH_MAP,
118     SLOVAK_MAP,
119     POLISH_MAP,
120     LATVIAN_MAP,
121     ARABIC_MAP,
122     LITHUANIAN_MAP,
123     SERBIAN_MAP,
124     AZERBAIJANI_MAP,
125     GEORGIAN_MAP,
126 ];
127
128 const Downcoder = {
129     'Initialize': function() {
130         if (Downcoder.map) { // already made
131             return;
132         }
133         Downcoder.map = {};
134         for (const lookup of ALL_DOWNCODE_MAPS) {
135             Object.assign(Downcoder.map, lookup);
136         }
137         Downcoder.regex = new RegExp(Object.keys(Downcoder.map).join('|'), 'g');
138     }
139 };
140
141 function downcode(slug) {
142     Downcoder.Initialize();
143     return slug.replace(Downcoder.regex, function(m) {
144         return Downcoder.map[m];
145     });
146 }
147
148
149 function URLify(s, num_chars, allowUnicode) {
150     // changes, e.g., "Petty theft" to "petty-theft"
151     if (!allowUnicode) {
152         s = downcode(s);
153     }
154     s = s.toLowerCase(); // convert to lowercase
155     // if downcode doesn't hit, the char will be stripped here
156     if (allowUnicode) {
157         // Keep Unicode letters including both lowercase and uppercase
158         // characters, whitespace, and dash; remove other characters.
159         s = XRegExp.replace(s, XRegExp('[^~\\p{L}\\p{N}]\\p{S}', 'g'), '');
160     } else {
161         s = s.replace(/[^\w\s]/g, ''); // remove unneeded chars
162     }
163     s = s.replace(/^(s+)$/g, ''); // trim leading/trailing spaces
164     s = s.replace(/[-\s]+/g, '-'); // convert spaces to hyphens
165     s = s.substring(0, num_chars); // trim to first num_chars chars
166     s = s.replace(/-+$/g, ''); // trim any trailing hyphens
167     return s;
168 }
169 window.URLify = URLify;
170

```

File: ./static/admin/js/jquery.init.js



```

52     const numCols = $this eq(1).children().length;
53     $parent.append('<tr class="' + options addCssClass + '"><td colspan="' + numCols + '"><a href="#">' + options addText + "</a></tr>");
54     addButton = $parent.find("tr:last a");
55 } else {
56     // Otherwise, insert it immediately after the last form:
57     $this filter(":last") after('<div class="' + options addCssClass + '"><a href="#">' + options addText + "</a></div>");
58     addButton = $this filter(":last") next().find("a");
59 }
60 }
61 addButton.on('click', addInlineClickHandler);
62 };
63
64 const addInlineClickHandler = function(e) {
65     e.preventDefault();
66     const template = $("#" + options prefix + "-empty");
67     const row = template.clone(true);
68     row.removeClass(options emptyCssClass);
69     .addClass(options formCssClass);
70     .attr("id", options prefix + "-" + nextIndex);
71     addInlineDeleteButton(row);
72     row.find("*").each(function() {
73         updateElementIndex(this, options prefix, totalForms val());
74     });
75     // Insert the new form when it has been fully edited.
76     row.insertBefore($ template);
77     // Update number of total forms.
78     $totalForms.val(parseInt(totalForms val()), 10) + 1);
79     nextIndex += 1;
80     // Hide the add button if there's a limit and it's been reached.
81     if ((maxForms val() !== "") && (maxForms val() - totalForms val()) <= 0) {
82         addButton.parent().hide();
83     }
84     // Show the remove buttons if there are more than min_num.
85     toggleDeleteButtonVisibility(row.closest('.inline-group'));
86
87     // Pass the new form to the post-add callback, if provided.
88     if (options added) {
89         options added(row);
90     }
91     $document.trigger('formset:added', [row, options prefix]);
92 };
93
94 /**
95  * The "X" button that is part of every unsaved inline.
96  * (When saved, it is replaced with a "Delete" checkbox.)
97  */
98 const addInlineDeleteButton = function(row) {
99     if (row.is("tr")) {
100         // If the forms are laid out in table rows, insert
101         // the remove button into the last table cell:
102         row.children(":last").append('<div><a class="' + options deleteCssClass + '" href="#">' + options deleteText + "</a></div>");
103     } else if (row.is("ul") || row.is("ol")) {
104         // If they're laid out as an ordered/unordered list,
105         // insert an <li> after the last list item:
106         row.append('<li><a class="' + options deleteCssClass + '" href="#">' + options deleteText + "</a></li>");
107     } else {
108         // Otherwise, just insert the remove button as the
109         // last child element of the form's container:
110         row.children(":first").append('<span><a class="' + options deleteCssClass + '" href="#">' + options deleteText + "</a></span>");
111     }
112     // Add delete handler for each row.
113     row.find("a." + options deleteCssClass).on('click', inlineDeleteHandler.bind(this));
114 };
115
116 const inlineDeleteHandler = function(e1) {
117     e1.preventDefault();
118     const deleteButton = $e1 target;
119     const row = deleteButton.closest('.form' + options formCssClass);
120     const inlineGroup = row.closest('.inline-group');
121     // Remove the parent form containing this button,
122     // and also remove the relevant row with non-field errors:
123     const prevRow = row.prev();
124     if (prevRow.length && prevRow.hasClass('row-form-errors')) {
125         prevRow.remove();
126     }
127     row.remove();
128     nextIndex -= 1;
129     // Pass the deleted form to the post-delete callback, if provided.
130     if (options removed) {
131         options removed(row);
132     }
133     $document.trigger('formset:removed', [row, options prefix]);
134     // Update the TOTAL_FORMS form count.
135     const forms = $("#" + options prefix + "-TOTAL_FORMS").val(forms.length);
136     // Show add button again once below maximum number.
137     if ((maxForms val() === "") || (maxForms val() - forms.length) > 0) {
138         addButton.parent().show();
139     }
140     // Hide the remove buttons if at min_num.
141     toggleDeleteButtonVisibility(inlineGroup);
142     // Also, update names and ids for all remaining form controls so
143     // they remain in sequence:
144     let i, formCount;
145     const updateElementCallback = function(i) {
146         updateElementIndex(this, options prefix, i);
147     };
148     for (i = 0, formCount = forms.length; i < formCount; i++) {
149         updateElementIndex($forms.get(i), options prefix, i);
150         $forms.get(i).find("*").each(updateElementCallback);
151     }
152 };
153
154 const toggleDeleteButtonVisibility = function(inlineGroup) {
155     if ((minForms val() !== "") && (minForms val() - totalForms val()) >= 0) {
156         inlineGroup.find('.inline-deletelink').hide();
157     } else {
158         inlineGroup.find('.inline-deletelink').show();
159     }
160 };
161
162 $this.each(function(i) {
163     $(this).not("." + options emptyCssClass).addClass(options formCssClass);
164 });
165
166

```

```

167 // Create the delete buttons for all unsaved inlines:
168 $this.filter('.' + options formCssClass + ':not(.has_original):not(.' + options emptyCssClass + ')') each(function() {
169     addInlineDeleteButton($(this));
170 });
171 toggleDeleteButtonVisibility($this);
172
173 // Create the add button, initially hidden.
174 addButton = options addButton;
175 addInlineAddButton();
176
177 // Show the add button if allowed to add more items.
178 // Note that max_num = None translates to a blank string.
179 const showAddButton = maxForms val() === " || (maxForms val() - totalForms val()) > 0;
180 if ($this.length && showAddButton) {
181     addButton parent().show();
182 } else {
183     addButton parent().hide();
184 }
185
186 return this;
187 };
188
189 /* Setup plugin defaults */
190 $fn formset defaults = {
191     prefix: "form", // The form prefix for your django formset
192     addText: "add another", // Text for the add link
193     deleteText: "remove", // Text for the delete link
194     addCssClass: "add-row", // CSS class applied to the add link
195     deleteCssClass: "delete-row", // CSS class applied to the delete link
196     emptyCssClass: "empty-row", // CSS class applied to the empty row
197     formCssClass: "dynamic-form", // CSS class applied to each form in a formset
198     added: null, // Function called each time a new form is added
199     removed: null, // Function called each time a form is deleted
200     addButton: null // Existing add button to use
201 };
202
203
204 // Tabular inlines -----
205 $fn tabularFormset = function(selector, options) {
206     const $rows = $(this);
207
208     const reinitDateTimeShortCuts = function() {
209         // Reinitialize the calendar and clock widgets by force
210         if (typeof DateTimeShortcuts !== "undefined") {
211             $(".datetimeshortcuts").remove();
212             DateTimeShortcuts.init();
213         }
214     };
215
216     const updateSelectFilter = function() {
217         // If any SelectFilter widgets are a part of the new form,
218         // instantiate a new SelectFilter instance for it.
219         if (typeof SelectFilter !== "undefined") {
220             $(".selectfilter").each(function(index, value) {
221                 SelectFilter.init(value.id, this.dataset.fieldName, false);
222             });
223             $(".selectfilterstacked").each(function(index, value) {
224                 SelectFilter.init(value.id, this.dataset.fieldName, true);
225             });
226         }
227     };
228
229     const initPrepopulatedFields = function(row) {
230         row.find('.prepopulated_field').each(function() {
231             const field = $(this);
232             input = field.find('input, select, textarea');
233             dependency_list = input.data('dependency_list') || [];
234             dependencies = [];
235             $ each dependency_list, function(i, field_name) {
236                 dependencies.push('#' + row.find('.field-' + field_name).find('input, select, textarea').attr('id'));
237             };
238             if (dependencies.length) {
239                 input.prepopulate(dependencies, input.attr('maxlength'));
240             }
241         });
242     };
243
244     $rows formset({
245         prefix: options.prefix,
246         addText: options.addText,
247         formCssClass: "dynamic-" + options.prefix,
248         deleteCssClass: "inline-deletelink",
249         deleteText: options.deleteText,
250         emptyCssClass: "empty-form",
251         added: function(row) {
252             initPrepopulatedFields(row);
253             reinitDateTimeShortCuts();
254             updateSelectFilter();
255         },
256         addButton: options.addButton
257     });
258
259     return $rows;
260 };
261
262 // Stacked inlines -----
263 $fn stackedFormset = function(selector, options) {
264     const $rows = $(this);
265     const updateInlineLabel = function(row) {
266         $ selector .find(".inline_label") each(function(i) {
267             const count = i + 1;
268             $(this).html($(this).html().replace(/(#d+)/g, "#" + count));
269         });
270     };
271
272     const reinitDateTimeShortCuts = function() {
273         // Reinitialize the calendar and clock widgets by force, yuck.
274         if (typeof DateTimeShortcuts !== "undefined") {
275             $(".datetimeshortcuts").remove();
276             DateTimeShortcuts.init();
277         }
278     };
279
280     const updateSelectFilter = function() {
281         // If any SelectFilter widgets were added, instantiate a new instance.

```

```

282     if (typeof SelectFilter !== "undefined") {
283         $(".selectfilter").each(function(index, value) {
284             SelectFilter.init(value.id, this.dataset.fieldName, false);
285         });
286         $(".selectfilterstacked").each(function(index, value) {
287             SelectFilter.init(value.id, this.dataset.fieldName, true);
288         });
289     }
290 };
291
292 const initPrepopulatedFields = function(row) {
293     row.find('.prepopulated_field').each(function() {
294         const field = $(this);
295         input = field.find('input, select, textarea');
296         dependency_list = input.data('dependency_list') || [];
297         dependencies = [];
298         $.each(dependency_list, function(i, field_name) {
299             dependencies.push('#' + row.find('.form-row .field-' + field_name).find('input, select, textarea').attr('id'));
300         });
301         if (dependencies.length) {
302             input.prepopulate(dependencies, input.attr('maxlength'));
303         }
304     });
305 };
306
307 $rows.formset({
308     prefix: options.prefix,
309     addText: options.addText,
310     formCssClass: "dynamic-" + options.prefix,
311     deleteCssClass: "inline-deletelink",
312     deleteText: options.deleteText,
313     emptyCssClass: "empty-form",
314     removed: updateInlineLabel,
315     added: function(row) {
316         initPrepopulatedFields(row);
317         reinitDateTimeShortCuts();
318         updateSelectFilter();
319         updateInlineLabel(row);
320     },
321     addButton: options.addButton
322 });
323
324 return $rows;
325 };
326
327 $(document).ready(function() {
328     $(".js-inline-admin-formset").each(function() {
329         const data = $(this).data();
330         inlineOptions = data.inlineFormset;
331         let selector;
332         switch(data.inlineType) {
333             case "stacked":
334                 selector = inlineOptions.name + "-group .inline-related";
335                 $selector = stackedFormset(selector, inlineOptions.options);
336                 break;
337             case "tabular":
338                 selector = inlineOptions.name + "-group .tabular.inline-related tbody:first > tr.form-row";
339                 $selector = tabularFormset(selector, inlineOptions.options);
340                 break;
341             }
342         });
343     });
344 }

```

**File:** ./static/admin/js/nav\_sidebar.js

```

1  'use strict';
2  {
3      const toggleNavSidebar = document.getElementById('toggle-nav-sidebar');
4      if (toggleNavSidebar !== null) {
5          const navLinks = document.querySelectorAll('#nav-sidebar a');
6          function disableNavLinkTabbing() {
7              for (const navLink of navLinks) {
8                  navLink.tabIndex = -1;
9              }
10         }
11         function enableNavLinkTabbing() {
12             for (const navLink of navLinks) {
13                 navLink.tabIndex = 0;
14             }
15         }
16
17         const main = document.getElementById('main');
18         let navSidebarsOpen = localStorage.getItem('django.admin.navSidebarsOpen');
19         if (navSidebarsOpen === null) {
20             navSidebarsOpen = 'true';
21         }
22         if (navSidebarsOpen === 'false') {
23             disableNavLinkTabbing();
24         }
25         main.classList.toggle('shifted', navSidebarsOpen === 'true');
26
27         toggleNavSidebar.addEventListener('click', function() {
28             if (navSidebarsOpen === 'true') {
29                 navSidebarsOpen = 'false';
30                 disableNavLinkTabbing();
31             } else {
32                 navSidebarsOpen = 'true';
33                 enableNavLinkTabbing();
34             }
35             localStorage.setItem('django.admin.navSidebarsOpen', navSidebarsOpen);
36             main.classList.toggle('shifted');
37         });
38     }
39
40     function initSidebarQuickFilter() {
41         const options = [];
42         const navSidebar = document.getElementById('nav-sidebar');
43         if (navSidebar) {
44             return;
45         }
46         navSidebar.querySelectorAll('th[scope=row] a').forEach((container) => {
47             options.push({title: container.innerHTML, node: container});
48         });
49
50         function checkValue(event) {
51             let filterValue = event.target.value;
52             if (filterValue) {
53                 filterValue = filterValue.toLowerCase();
54             }
55             if (event.key === 'Escape') {
56                 filterValue = '';
57                 event.target.value = ''; // clear input
58             }
59             let matches = false;
60             for (const o of options) {
61                 let displayValue = '';
62                 if (filterValue) {
63                     if (o.title.toLowerCase().indexOf(filterValue) === -1) {
64                         displayValue = 'none';
65                     } else {
66                         matches = true;
67                     }
68                 }
69                 // show/hide parent <TR>
70                 o.node.parentNode.parentNode.style.display = displayValue;
71             }
72             if (filterValue || matches) {
73                 event.target.classList.remove('no-results');
74             } else {
75                 event.target.classList.add('no-results');
76             }
77             sessionStorage.setItem('django.admin.navSidebarFilterValue', filterValue);
78         }
79
80         const nav = document.getElementById('nav-filter');
81         nav.addEventListener('change', checkValue, false);
82         nav.addEventListener('input', checkValue, false);
83         nav.addEventListener('keyup', checkValue, false);
84
85         const storedValue = sessionStorage.getItem('django.admin.navSidebarFilterValue');
86         if (storedValue) {
87             nav.value = storedValue;
88             checkValue({target: nav, key: ''});
89         }
90     }
91     window.initSidebarQuickFilter = initSidebarQuickFilter;
92     initSidebarQuickFilter();
93 }

```

#### File: ./static/admin/js/core.js

```

1  // Core JavaScript helper functions
2  'use strict';
3
4  // quickElement(tagType, parentReference [, textInChildNode, attribute, attributeValue ...]);
5  function quickElement() {
6      const obj = document.createElement(arguments[0]);
7      if (arguments[2]) {
8          const textNode = document.createTextNode(arguments[2]);
9          obj.appendChild(textNode);
10     }
11     const len = arguments.length;
12     for (let i = 3, i < len; i += 2) {
13         obj.setAttribute(arguments[i], arguments[i + 1]);
14     }
15     arguments[1].appendChild(obj);
16     return obj;

```



```

17 }
18
19 // "a" is reference to an object
20 function removeChildren(a) {
21   while (a.hasChildNodes()) {
22     a.removeChild(a.lastChild);
23   }
24 }
25
26 // -----
27 // Find-position functions by PPK
28 // See https://www.quirksmode.org/js/findpos.html
29 // -----
30 function findPosX(obj) {
31   let curleft = 0;
32   if (obj.offsetParent) {
33     while (obj.offsetParent) {
34       curleft += obj.offsetLeft - obj.scrollLeft;
35       obj = obj.offsetParent;
36     }
37   } else if (obj.x) {
38     curleft += obj.x;
39   }
40   return curleft;
41 }
42
43 function findPosY(obj) {
44   let curtop = 0;
45   if (obj.offsetParent) {
46     while (obj.offsetParent) {
47       curtop += obj.offsetTop - obj.scrollTop;
48       obj = obj.offsetParent;
49     }
50   } else if (obj.y) {
51     curtop += obj.y;
52   }
53   return curtop;
54 }
55
56 // -----
57 // Date object extensions
58 // -----
59 {
60   Date.prototype.getTwelveHours = function() {
61     return this.getHours() % 12 || 12;
62   };
63
64   Date.prototype.getTwoDigitMonth = function() {
65     return (this.getMonth() < 9) ? '0' + (this.getMonth() + 1) : (this.getMonth() + 1);
66   };
67
68   Date.prototype.getTwoDigitDate = function() {
69     return (this.getDate() < 10) ? '0' + this.getDate() : this.getDate();
70   };
71
72   Date.prototype.getTwoDigitTwelveHour = function() {
73     return (this.getTwelveHours() < 10) ? '0' + this.getTwelveHours() : this.getTwelveHours();
74   };
75
76   Date.prototype.getTwoDigitHour = function() {
77     return (this.getHours() < 10) ? '0' + this.getHours() : this.getHours();
78   };
79
80   Date.prototype.getTwoDigitMinute = function() {
81     return (this.getMinutes() < 10) ? '0' + this.getMinutes() : this.getMinutes();
82   };
83
84   Date.prototype.getTwoDigitSecond = function() {
85     return (this.getSeconds() < 10) ? '0' + this.getSeconds() : this.getSeconds();
86   };
87
88   Date.prototype.getAbbrevMonthName = function() {
89     return typeof window.CalendarNamespace === "undefined"
90       ? this.getTwoDigitMonth()
91       : window.CalendarNamespace.monthsOfYearAbbrev[this.getMonth()];
92   };
93
94   Date.prototype.getFullMonthName = function() {
95     return typeof window.CalendarNamespace === "undefined"
96       ? this.getTwoDigitMonth()
97       : window.CalendarNamespace.monthsOfYear[this.getMonth()];
98   };
99
100   Date.prototype.strftime = function(format) {
101     const fields = {
102       b: this.getAbbrevMonthName(),
103       B: this.getFullMonthName(),
104       c: this.toString(),
105       d: this.getTwoDigitDate(),
106       H: this.getTwoDigitHour(),
107       I: this.getTwoDigitTwelveHour(),
108       m: this.getTwoDigitMonth(),
109       M: this.getTwoDigitMinute(),
110       p: (this.getHours() >= 12) ? 'PM' : 'AM',
111       S: this.getTwoDigitSecond(),
112       w: '0' + this.getDay(),
113       x: this.toLocaleDateString(),
114       X: this.toLocaleTimeString(),
115       y: (" " + this.getFullYear()).substr(2, 4),
116       Y: " " + this.getFullYear(),
117       '%': '%',
118     };
119     let result = "", i = 0;
120     while (i < format.length) {
121       if (format.charAt(i) === '%') {
122         result = result + fields[format.charAt(i + 1)];
123         ++i;
124       } else {
125         result = result + format.charAt(i);
126       }
127       ++i;
128     }
129     return result;
130   };
131 }

```

```

132
133 // -----
134 // String object extensions
135 // -----
136 String.prototype.strptime = function(format) {
137     const split_format = format.split(/[\/-]/);
138     const date = this.split(/[\/-]/);
139     let i = 0;
140     let day, month, year;
141     while (i < split_format.length) {
142         switch (split_format[i]) {
143             case "%d":
144                 day = date[i];
145                 break;
146             case "%m":
147                 month = date[i] - 1;
148                 break;
149             case "%y":
150                 year = date[i];
151                 break;
152             case "%Y":
153                 // A %y value in the range of [00, 68] is in the current
154                 // century, while [69, 99] is in the previous century,
155                 // according to the Open Group Specification.
156                 if (parseInt(date[i], 10) >= 69) {
157                     year = date[i];
158                 } else {
159                     year = (new Date(Date.UTC(date[i], 0))).getUTCFullYear() + 100;
160                 }
161                 break;
162             }
163             ++i;
164         }
165         // Create Date object from UTC since the parsed value is supposed to be
166         // in UTC, not local time. Also, the calendar uses UTC functions for
167         // date extraction.
168         return new Date(Date.UTC(year, month, day));
169     };
170 }

```

#### File: ./static/admin/js/change\_form.js

```

1 'use strict';
2 {
3     const inputTags = ['BUTTON', 'INPUT', 'SELECT', 'TEXTAREA'];
4     const modelName = document.getElementById('django-admin-form-add-constants').dataset.modelName;
5     if (modelName) {
6         const form = document.getElementById(modelName + '_form');
7         for (const element of form.elements) {
8             // HTMLElement.offsetParent returns null when the element is not
9             // rendered.
10            if (inputTags.includes(element.tagName) && !element.disabled && element.offsetParent) {
11                element.focus();
12                break;
13            }
14        }
15    }
16 }

```

#### File: ./static/admin/js/popup\_response.js

```

1 /*global opener */
2 'use strict';
3 {
4     const initData = JSON.parse(document.getElementById('django-admin-popup-response-constants').dataset.popupResponse);
5     switch (initData.action) {
6         case 'change':
7             opener.dismissChangeRelatedObjectPopup(window, initData.value, initData.obj, initData.new_value);
8             break;
9         case 'delete':
10            opener.dismissDeleteRelatedObjectPopup(window, initData.value);
11            break;
12         default:
13             opener.dismissAddRelatedObjectPopup(window, initData.value, initData.obj);
14             break;
15     }
16 }

```

#### File: ./static/admin/js/calendar.js

```

1 /*global gettext, pgettext, get_format, quickElement, removeChildren*/
2 /*
3  calendar.js - Calendar functions by Adrian Holovaty
4  depends on core.js for utility functions like removeChildren or quickElement
5  */
6 'use strict';
7 {
8     // CalendarNamespace -- Provides a collection of HTML calendar-related helper functions
9     const CalendarNamespace = {
10         monthsOfYear: [
11             gettext('January'),
12             gettext('February'),
13             gettext('March'),
14             gettext('April'),
15             gettext('May'),
16             gettext('June'),
17             gettext('July'),
18             gettext('August'),
19             gettext('September'),
20             gettext('October'),
21             gettext('November'),
22             gettext('December')
23         ],
24         monthsOfYearAbbrev: [
25             pgettext('abbrev. month January', 'Jan'),

```

```

1 pgettext('abbrev. month February', 'Feb'),
2 pgettext('abbrev. month March', 'Mar'),
3 pgettext('abbrev. month April', 'Apr'),
4 pgettext('abbrev. month May', 'May'),
5 pgettext('abbrev. month June', 'Jun'),
6 pgettext('abbrev. month July', 'Jul'),
7 pgettext('abbrev. month August', 'Aug'),
8 pgettext('abbrev. month September', 'Sep'),
9 pgettext('abbrev. month October', 'Oct'),
10 pgettext('abbrev. month November', 'Nov'),
11 pgettext('abbrev. month December', 'Dec')
12 ],
13 daysOfWeek: [
14 pgettext('one letter Sunday', 'S'),
15 pgettext('one letter Monday', 'M'),
16 pgettext('one letter Tuesday', 'T'),
17 pgettext('one letter Wednesday', 'W'),
18 pgettext('one letter Thursday', 'T'),
19 pgettext('one letter Friday', 'F'),
20 pgettext('one letter Saturday', 'S')
21 ],
22 firstDayOfWeek: parseInt( get_format('FIRST_DAY_OF_WEEK') ),
23 isLeapYear: function( year ) {
24     return ((year % 4) === 0 && ((year % 100) !== 0) || ((year % 400) === 0));
25 },
26 getDaysInMonth: function( month, year ) {
27     let days;
28     if ( month === 1 || month === 3 || month === 5 || month === 7 || month === 8 || month === 10 || month === 12 ) {
29         days = 31;
30     }
31     else if ( month === 4 || month === 6 || month === 9 || month === 11 ) {
32         days = 30;
33     }
34     else if ( month === 2 && CalendarNamespace.isLeapYear( year ) ) {
35         days = 29;
36     }
37     else {
38         days = 28;
39     }
40     return days;
41 },
42 draw: function( month, year, div_id, callback, selected ) { // month = 1-12, year = 1-9999
43     const today = new Date();
44     const todayDay = today.getDate();
45     const todayMonth = today.getMonth() + 1;
46     const todayYear = today.getFullYear();
47     let todayClass = "";
48
49     // Use UTC functions here because the date field does not contain time
50     // and using the UTC function variants prevent the local time offset
51     // from altering the date, specifically the day field. For example:
52     //
53     // var x = new Date('2013-10-02');
54     // var day = x.getDate();
55     //
56     // The day variable above will be 1 instead of 2 in, say, US Pacific time
57     // zone.
58     let isSelectedMonth = false;
59     if ( typeof selected !== 'undefined' ) {
60         isSelectedMonth = (selected.getUTCFullYear() === year && (selected.getUTCMonth() + 1) === month);
61     }
62
63     month = parseInt( month );
64     year = parseInt( year );
65     const calDiv = document.getElementById( div_id );
66     removeChildren( calDiv );
67     const calTable = document.createElement( 'table' );
68     quickElement( 'caption', calTable, CalendarNamespace.monthsOfYear( month - 1 ) + ' ' + year );
69     const tableBody = quickElement( 'tbody', calTable );
70
71     // Draw days-of-week header
72     let tableRow = quickElement( 'tr', tableBody );
73     for ( let i = 0; i < 7; i++ ) {
74         quickElement( 'th', tableRow, CalendarNamespace.daysOfWeek[ (i + CalendarNamespace.firstDayOfWeek) % 7 ] );
75     }
76
77     const startingPos = new Date( year, month - 1, 1 - CalendarNamespace.firstDayOfWeek ).getDay();
78     const days = CalendarNamespace.getDaysInMonth( month, year );
79
80     let nonDayCell;
81
82     // Draw blanks before first of month
83     tableRow = quickElement( 'tr', tableBody );
84     for ( let i = 0; i < startingPos; i++ ) {
85         nonDayCell = quickElement( 'td', tableRow, ' ' );
86         nonDayCell.className = "nonday";
87     }
88
89     function calendarMonth( y, m ) {
90         function onClick( e ) {
91             e.preventDefault();
92             callback( y, m, this.textContent );
93         }
94         return onClick;
95     }
96
97     // Draw days of month
98     let currentDay = 1;
99     for ( let i = startingPos; currentDay <= days; i++ ) {
100         if ( i % 7 === 0 && currentDay !== 1 ) {
101             tableRow = quickElement( 'tr', tableBody );
102         }
103         if ( (currentDay === todayDay) && (month === todayMonth) && (year === todayYear) ) {
104             todayClass = 'today';
105         }
106         else {
107             todayClass = "";
108         }
109
110         // use UTC function; see above for explanation.
111         if ( isSelectedMonth && currentDay === selected.getUTCDate() ) {
112             if ( todayClass !== "" ) {
113                 todayClass += " ";
114             }
115         }
116     }
117 }

```

```

141         todayClass += "selected";
142     }
143
144     const cell = quickElement('td', tableRow, '', 'class', todayClass);
145     const link = quickElement('a', cell, currentDay, 'href', '#');
146     link.addEventListener('click', calendarMonth(year, month));
147     currentDay++;
148 }
149
150 // Draw blanks after end of month (optional, but makes for valid code)
151 while (tableRow.childNodes.length < 7) {
152     nonDayCell = quickElement('td', tableRow, '');
153     nonDayCell.className = "nonday";
154 }
155
156 calDiv.appendChild(calTable);
157 }
158 };
159
160 // Calendar -- A calendar instance
161 function Calendar(div_id, callback, selected) {
162     // div_id (string) is the ID of the element in which the calendar will
163     // be displayed
164     // callback (string) is the name of a JavaScript function that will be
165     // called with the parameters (year, month, day) when a day in the
166     // calendar is clicked
167     this.div_id = div_id;
168     this.callback = callback;
169     this.today = new Date();
170     this.currentMonth = this.today.getMonth() + 1;
171     this.currentYear = this.today.getFullYear();
172     if (typeof selected !== 'undefined') {
173         this.selected = selected;
174     }
175 }
176
177 Calendar.prototype = {
178     drawCurrent: function() {
179         CalendarNamespace.draw(this.currentMonth, this.currentYear, this.div_id, this.callback, this.selected);
180     },
181     drawDate: function(month, year, selected) {
182         this.currentMonth = month;
183         this.currentYear = year;
184
185         if(selected) {
186             this.selected = selected;
187         }
188
189         this.drawCurrent();
190     },
191     drawPreviousMonth: function() {
192         if (this.currentMonth === 1) {
193             this.currentMonth = 12;
194             this.currentYear--;
195         }
196         else {
197             this.currentMonth--;
198         }
199         this.drawCurrent();
200     },
201     drawNextMonth: function() {
202         if (this.currentMonth === 12) {
203             this.currentMonth = 1;
204             this.currentYear++;
205         }
206         else {
207             this.currentMonth++;
208         }
209         this.drawCurrent();
210     },
211     drawPreviousYear: function() {
212         this.currentYear--;
213         this.drawCurrent();
214     },
215     drawNextYear: function() {
216         this.currentYear++;
217         this.drawCurrent();
218     }
219 };
220 window.Calendar = Calendar;
221 window.CalendarNamespace = CalendarNamespace;

```

**File: ./static/admin/js/collapse.js**

```

1  /*global gettext*/
2  'use strict';
3  {
4    window.addEventListener('load', function() {
5      // Add anchor tag for Show/Hide link
6      const fieldsets = document.querySelectorAll('fieldset.collapse');
7      for (const (i, elem) of fieldsets.entries()) {
8        // Don't hide if fields in this fieldset have errors
9        if (elem.querySelectorAll('div.errors, ul.errorlist').length === 0) {
10          elem.classList.add('collapsed');
11          const h2 = elem.querySelector('h2');
12          const link = document.createElement('a');
13          link.id = 'fieldsetcollapser' + i;
14          link.className = 'collapse-toggle';
15          link.href = '#';
16          link.textContent = gettext('Show');
17          h2.appendChild(document.createTextNode(' '));
18          h2.appendChild(link);
19          h2.appendChild(document.createTextNode(' '));
20        }
21      }
22      // Add toggle to hide/show anchor tag
23      const toggleFunc = function(ev) {
24        if (ev.target.matches('.collapse-toggle')) {
25          ev.preventDefault();
26          ev.stopPropagation();
27          const fieldset = ev.target.closest('fieldset');
28          if (fieldset.classList.contains('collapsed')) {
29            // Show
30            ev.target.textContent = gettext('Hide');
31            fieldset.classList.remove('collapsed');
32          } else {
33            // Hide
34            ev.target.textContent = gettext('Show');
35            fieldset.classList.add('collapsed');
36          }
37        }
38      };
39      document.querySelectorAll('fieldset.module').forEach(function(el) {
40        el.addEventListener('click', toggleFunc);
41      });
42    });
43 }

```

**File:** ./static/admin/js/SelectBox.js

```

1  'use strict';
2  {
3      const SelectBox = {
4          cache: {},
5          init: function(id) {
6              const box = document.getElementById(id);
7              SelectBox.cache[id] = [];
8              const cache = SelectBox.cache[id];
9              for (const node of box.options) {
10                 cache.push({value: node.value, text: node.text, displayed: 1});
11             }
12         },
13         redisplay: function(id) {
14             // Repopulate HTML select box from cache
15             const box = document.getElementById(id);
16             const scroll_value_from_top = box.scrollTop;
17             box.innerHTML = "";
18             for (const node of SelectBox.cache[id]) {
19                 if (node.displayed) {
20                     const new_option = new Option(node.text, node.value, false, false);
21                     // Shows a tooltip when hovering over the option
22                     new_option.title = node.text;
23                     box.appendChild(new_option);
24                 }
25             }
26             box.scrollTop = scroll_value_from_top;
27         },
28         filter: function(id, text) {
29             // Redisplay the HTML select box, displaying only the choices containing ALL
30             // the words in text. (It's an AND search.)
31             const tokens = text.toLowerCase().split(/\s+/);
32             for (const node of SelectBox.cache[id]) {
33                 node.displayed = 1;
34                 const node_text = node.text.toLowerCase();
35                 for (const token of tokens) {
36                     if (!node_text.includes(token)) {
37                         node.displayed = 0;
38                         break; // Once the first token isn't found we're done
39                     }
40                 }
41             }
42             SelectBox.redisplay(id);
43         },
44         delete_from_cache: function(id, value) {
45             let delete_index = null;
46             const cache = SelectBox.cache[id];
47             for (const [i, node] of cache.entries()) {
48                 if (node.value === value) {
49                     delete_index = i;
50                     break;
51                 }
52             }
53             cache.splice(delete_index, 1);
54         },
55         add_to_cache: function(id, option) {
56             SelectBox.cache[id].push({value: option.value, text: option.text, displayed: 1});
57         },
58         cache_contains: function(id, value) {
59             // Check if an item is contained in the cache
60             for (const node of SelectBox.cache[id]) {
61                 if (node.value === value) {
62                     return true;
63                 }
64             }
65             return false;
66         },
67         move: function(from, to) {
68             const from_box = document.getElementById(from);
69             for (const option of from_box.options) {
70                 const option_value = option.value;
71                 if (option.selected && SelectBox.cache_contains(from, option_value)) {
72                     SelectBox.add_to_cache(to, {value: option_value, text: option.text, displayed: 1});
73                     SelectBox.delete_from_cache(from, option_value);
74                 }
75             }
76             SelectBox.redisplay(from);
77             SelectBox.redisplay(to);
78         },
79         move_all: function(from, to) {
80             const from_box = document.getElementById(from);
81             for (const option of from_box.options) {
82                 const option_value = option.value;
83                 if (SelectBox.cache_contains(from, option_value)) {
84                     SelectBox.add_to_cache(to, {value: option_value, text: option.text, displayed: 1});
85                     SelectBox.delete_from_cache(from, option_value);
86                 }
87             }
88             SelectBox.redisplay(from);
89             SelectBox.redisplay(to);
90         },
91         sort: function(id) {
92             SelectBox.cache[id].sort(function(a, b) {
93                 a = a.text.toLowerCase();
94                 b = b.text.toLowerCase();
95                 if (a > b) {
96                     return 1;
97                 }
98                 if (a < b) {
99                     return -1;
100                }
101                return 0;
102            });
103        },
104        select_all: function(id) {
105            const box = document.getElementById(id);
106            for (const option of box.options) {
107                option.selected = true;
108            }
109        }
110    };
111    window.SelectBox = SelectBox;
112 }

```

**File: ./static/admin/js/vendor/select2/i18n/sr.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var n=jQuery.fn.select2 amd n define!["select2/i18n/sr",[],function() {function n(n,e,r,t) {return n%10==1&&n%100!=1
```

**File: ./static/admin/js/vendor/select2/i18n/es.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var e=jQuery.fn.select2 amd e define!["select2/i18n/es",[],function() {return(errorLoading,function() {return"No se puc
```

**File: ./static/admin/js/vendor/select2/i18n/fa.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var n=jQuery.fn.select2 amd n define!["select2/i18n/fa",[],function() {return(errorLoading,function() {return" ".input
```

**File: ./static/admin/js/vendor/select2/i18n/nb.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var e=jQuery.fn.select2 amd e define!["select2/i18n/nb",[],function() {return(errorLoading,function() {return"Kunne ikk
```

**File: ./static/admin/js/vendor/select2/i18n/nl.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var e=jQuery.fn.select2 amd e define!["select2/i18n/nl",[],function() {return(errorLoading,function() {return"De resulta
```

**File: ./static/admin/js/vendor/select2/i18n/ne.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var n=jQuery.fn.select2 amd n define!["select2/i18n/ne",[],function() {return(errorLoading,function() {return" ".input'
```

**File: ./static/admin/js/vendor/select2/i18n/ja.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var n=jQuery.fn.select2 amd n define!["select2/i18n/ja",[],function() {return(errorLoading,function() {return""}.inputTc
```

**File: ./static/admin/js/vendor/select2/i18n/hr.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var n=jQuery.fn.select2 amd n define!["select2/i18n/hr",[],function() {function n(n) {var e=" "+n+" znak".return n%10
```

**File: ./static/admin/js/vendor/select2/i18n/uk.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var n=jQuery.fn.select2 amd n define!["select2/i18n/uk",[],function() {function n(n,e,u,r) {return n%100>10&&n%100<
```

**File: ./static/admin/js/vendor/select2/i18n/eu.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var e=jQuery.fn.select2 amd e define!["select2/i18n/eu",[],function() {return(inputTooLong,function(e) {var t=e.input l
```

**File: ./static/admin/js/vendor/select2/i18n/tk.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var e=jQuery.fn.select2 amd e define!["select2/i18n/tk",[],function() {return(errorLoading,function() {return"Netije kler
```

**File: ./static/admin/js/vendor/select2/i18n/et.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var e=jQuery.fn.select2 amd e define!["select2/i18n/et",[],function() {return(inputTooLong,function(e) {var n=e.input l
```

**File: ./static/admin/js/vendor/select2/i18n/pt.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2 amd var e=jQuery.fn.select2 amd e define!["select2/i18n/pt",[],function() {return(errorLoading,function() {return"Os resulta
```

**File: ./static/admin/js/vendor/select2/i18n/fi.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var n=jQuery fn select2 amd n define("select2/i18n/az",[],function() {return {errorLoading:function() {return "Tuloksia ei
```

**File: ./static/admin/js/vendor/select2/i18n/az.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var n=jQuery fn select2 amd n define("select2/i18n/az",[],function() {return {inputTooLong:function(n) {return n input
```

**File: ./static/admin/js/vendor/select2/i18n/ca.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var e=jQuery fn select2 amd e define("select2/i18n/ca",[],function() {return {errorLoading:function() {return "La crrega
```

**File: ./static/admin/js/vendor/select2/i18n/bg.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var n=jQuery fn select2 amd n define("select2/i18n/bg",[],function() {return {inputTooLong:function(n) {var e=n input
```

**File: ./static/admin/js/vendor/select2/i18n/mk.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var n=jQuery fn select2 amd n define("select2/i18n/mk",[],function() {return {inputTooLong:function(n) {var e=(n input
```

**File: ./static/admin/js/vendor/select2/i18n/is.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var n=jQuery fn select2 amd n define("select2/i18n/is",[],function() {return {inputTooLong:function(n) {var t=n input le
```

**File: ./static/admin/js/vendor/select2/i18n/it.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var e=jQuery fn select2 amd e define("select2/i18n/it",[],function() {return {errorLoading:function() {return "I risultati n
```

**File: ./static/admin/js/vendor/select2/i18n/id.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var n=jQuery fn select2 amd n define("select2/i18n/id",[],function() {return {errorLoading:function() {return "Data tidak
```

**File: ./static/admin/js/vendor/select2/i18n/hi.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var n=jQuery fn select2 amd n define("select2/i18n/hi",[],function() {return {errorLoading:function() {return " " },inp
```

**File: ./static/admin/js/vendor/select2/i18n/sr-Cyrl.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var n=jQuery fn select2 amd n define("select2/i18n/sr-Cyrl",[],function() {function n(n,e,r,u) {return n%10==1&&n%10
```

**File: ./static/admin/js/vendor/select2/i18n/af.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var e=jQuery fn select2 amd e define("select2/i18n/af",[],function() {return {errorLoading:function() {return "Die result
```

**File: ./static/admin/js/vendor/select2/i18n/zh-CN.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var n=jQuery fn select2 amd n define("select2/i18n/zh-CN",[],function() {return {errorLoading:function() {return "" },inp
```

**File: ./static/admin/js/vendor/select2/i18n/lv.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var e=jQuery fn select2 amd e define("select2/i18n/lv",[],function() {function e(e,n,u,i) {return 11===e'n e'%10==1?
```

**File: ./static/admin/js/vendor/select2/i18n/sk.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn&&jQuery fn select2&&jQuery fn select2 amd)var e=jQuery fn select2 amd e define("select2/i18n/sk",[],function() {var e=(2,function(e) {return e?"dva":"dve"},3,fu
```



**File: ./static/admin/js/vendor/select2/i18n/de.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var e=jQuery.fn.select2.amd,e.define("select2/i18n/de",[],function() {return({errorLoading:function() {return"Die Ergeb
```

**File: ./static/admin/js/vendor/select2/i18n/ar.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd.n.define("select2/i18n/ar",[],function() {return({errorLoading:function() {return" ".input
```

**File: ./static/admin/js/vendor/select2/i18n/gl.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var e=jQuery.fn.select2.amd.e.define("select2/i18n/gl",[],function() {return({errorLoading:function() {return"Non foi po
```

**File: ./static/admin/js/vendor/select2/i18n/sl.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var e=jQuery.fn.select2.amd.e.define("select2/i18n/sl",[],function() {return({errorLoading:function() {return"Zadetkov
```

**File: ./static/admin/js/vendor/select2/i18n/sq.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var e=jQuery.fn.select2.amd.e.define("select2/i18n/sq",[],function() {return({errorLoading:function() {return"Rezultate
```

**File: ./static/admin/js/vendor/select2/i18n/he.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd.n.define("select2/i18n/he",[],function() {return({errorLoading:function() {return" ".input
```

**File: ./static/admin/js/vendor/select2/i18n/ps.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd.n.define("select2/i18n/ps",[],function() {return({errorLoading:function() {return" ".input
```

**File: ./static/admin/js/vendor/select2/i18n/sv.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd.n.define("select2/i18n/sv",[],function() {return({errorLoading:function() {return"Resultat k
```

**File: ./static/admin/js/vendor/select2/i18n/bn.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd.n.define("select2/i18n/bn",[],function() {return({errorLoading:function() {return" ".input
```

**File: ./static/admin/js/vendor/select2/i18n/hsb.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd.n.define("select2/i18n/hsb",[],function() {var n=["znamjeko","znamjece","znamjeka","zn
```

**File: ./static/admin/js/vendor/select2/i18n/vi.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd.n.define("select2/i18n/vi",[],function() {return({inputTooLong:function(n) {return"Vui lng ;
```

**File: ./static/admin/js/vendor/select2/i18n/km.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd.n.define("select2/i18n/km",[],function() {return({errorLoading:function() {return""}).input
```

**File: ./static/admin/js/vendor/select2/i18n/ms.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 :function() {if(jQuery&jQuery.fn&jQuery.fn.select2&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd.n.define("select2/i18n/ms",[],function() {return({errorLoading:function() {return"Keputuse
```

**File: ./static/admin/js/vendor/select2/i18n/fr.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var e=jQuery fn select2 amd e define("select2/i18n/fr",[],function() {return {errorLoading:function() {return "Les rsultat
```

#### File: ./static/admin/js/vendor/select2/i18n/cs.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var e=jQuery fn select2 amd e define("select2/i18n/cs",[],function() {function e(e,n){switch(e){case 2 return n?"dva":
```

#### File: ./static/admin/js/vendor/select2/i18n/hu.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var e=jQuery fn select2 amd e define("select2/i18n/hu",[],function() {return {errorLoading:function() {return "Az eredm
```

#### File: ./static/admin/js/vendor/select2/i18n/ro.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var e=jQuery fn select2 amd e define("select2/i18n/ro",[],function() {return {errorLoading:function() {return "Rezultate
```

#### File: ./static/admin/js/vendor/select2/i18n/pl.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var n=jQuery fn select2 amd n define("select2/i18n/pl",[],function() {var n=["znak","znaki","znakw"],e=["element","e
```

#### File: ./static/admin/js/vendor/select2/i18n/da.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var e=jQuery fn select2 amd e define("select2/i18n/da",[],function() {return {errorLoading:function() {return "Resultate
```

#### File: ./static/admin/js/vendor/select2/i18n/dsb.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var n=jQuery fn select2 amd n define("select2/i18n/dsb",[],function() {var n=["znamuko","znamuce","znamuka","znai
```

#### File: ./static/admin/js/vendor/select2/i18n/pt-BR.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var e=jQuery fn select2 amd e define("select2/i18n/pt-BR",[],function() {return {errorLoading:function() {return "Os res
```

#### File: ./static/admin/js/vendor/select2/i18n/ko.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var n=jQuery fn select2 amd n define("select2/i18n/ko",[],function() {return {errorLoading:function() {return " .").input
```

#### File: ./static/admin/js/vendor/select2/i18n/ru.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var n=jQuery fn select2 amd n define("select2/i18n/ru",[],function() {function n(n,e,r,u) {return n%10<5&&n%10>0&&e
```

#### File: ./static/admin/js/vendor/select2/i18n/en.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var e=jQuery fn select2 amd e define("select2/i18n/en",[],function() {return {errorLoading:function() {return "The resul
```

#### File: ./static/admin/js/vendor/select2/i18n/el.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var n=jQuery fn select2 amd n define("select2/i18n/el",[],function() {return {errorLoading:function() {return " .").inp
```

#### File: ./static/admin/js/vendor/select2/i18n/th.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var n=jQuery fn select2 amd n define("select2/i18n/th",[],function() {return {errorLoading:function() {return ""}.inputTc
```

#### File: ./static/admin/js/vendor/select2/i18n/tr.js

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery$&jQuery fn$&jQuery fn select2$&jQuery fn select2 amd|var n=jQuery fn select2 amd n define("select2/i18n/tr",[],function() {return {errorLoading:function() {return "Sonu ykler
```

**File: ./static/admin/js/vendor/select2/i18n/ka.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery&&jQuery.fn&&jQuery.fn.select2&&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd,n.define("select2/i18n/ka",[],function(){return(errorLoading:function(){return" "}).input
```

**File: ./static/admin/js/vendor/select2/i18n/lt.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery&&jQuery.fn&&jQuery.fn.select2&&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd,n.define("select2/i18n/lt",[],function(){function n(n,e,i,t){return n%10==1&&(n%100<1
```

**File: ./static/admin/js/vendor/select2/i18n/zh-TW.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery&&jQuery.fn&&jQuery.fn.select2&&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd,n.define("select2/i18n/zh-TW",[],function(){return(InputTooLong:function(n){return""+(
```

**File: ./static/admin/js/vendor/select2/i18n/bs.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery&&jQuery.fn&&jQuery.fn.select2&&jQuery.fn.select2.amd)var e=jQuery.fn.select2.amd,e.define("select2/i18n/bs",[],function(){function e(e,n,r,t){return e%10==1&&(e%100!=
```

**File: ./static/admin/js/vendor/select2/i18n/hy.js**

```
1 /*! Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2
3 (function() {if(jQuery&&jQuery.fn&&jQuery.fn.select2&&jQuery.fn.select2.amd)var n=jQuery.fn.select2.amd,n.define("select2/i18n/hy",[],function(){return(errorLoading:function(){return" "}).input
```

**File: ./static/admin/js/vendor/select2/select2.full.js**

```
1 /*!
2  * Select2 4.0.13
3  * https://select2.github.io
4  *
5  * Released under the MIT license
6  * https://github.com/select2/select2/blob/master/LICENSE.md
7  */
8 (function (factory) {
9   if (typeof define === 'function' && define.amd) {
10     // AMD. Register as an anonymous module.
11     define(['jquery'], factory);
12   } else if (typeof module === 'object' && module.exports) {
13     // Node/CommonJS
14     module.exports = function (root, jQuery) {
15       if (jQuery === undefined) {
16         // require('jquery') returns a factory that requires window to
17         // build a jQuery instance, we normalize how we use modules
18         // that require this pattern but the window provided is a noop
19         // if it's defined (how jquery works)
20         if (typeof window !== 'undefined') {
21           jQuery = require('jquery');
22         }
23         else {
24           jQuery = require('jquery')(root);
25         }
26       }
27       factory(jQuery);
28       return jQuery;
29     };
30   } else {
31     // Browser globals
32     factory(jQuery);
33   }
34 } (function (jQuery) {
35   // This is needed so we can catch the AMD loader configuration and use it
36   // The inner file should be wrapped (by 'banner.start.js') in a function that
37   // returns the AMD loader references.
38   var S2 =(function () {
39     // Restore the Select2 AMD loader so it can be used
40     // Needed mostly in the language files, where the loader is not inserted
41     if (jQuery && jQuery.fn && jQuery.fn.select2 && jQuery.fn.select2.amd) {
42       var S2 = jQuery.fn.select2.amd;
43     }
44     var S2 =(function () { if (!S2 || !S2.requirejs) {
45       if (!S2) { S2 = {}; } else { require = S2; }
46       /**
47        * @license almond 0.3.3 Copyright jQuery Foundation and other contributors.
48        * Released under MIT license, http://github.com/requirejs/almond/LICENSE
49        */
50        //Going sloppy to avoid 'use strict' string cost, but strict practices should
51        //be followed.
52        /*global setTimeout: false */
53
54        var requirejs = require;
55        (function (undef) {
56          var main, req, makeMap, handlers;
57          defined = {},
58            waiting = {},
59            config = {},
60            defining = {},
61            hasOwn = Object.prototype.hasOwnProperty,
62            aps = [].slice,
63            jsSuffixRegExp = /\.js$/;
64
65          function hasProp(obj, prop) {
66            return hasOwn.call(obj, prop);
67          }
68
69          /**
70           * Given a relative module name, like ./something, normalize it to
```

```

71  * a real name that can be mapped to a path.
72  * @param {String} name the relative name
73  * @param {String} baseName a real name that the name arg is relative
74  * to.
75  * @returns {String} normalized name
76  */
77  function normalize(name, baseName) {
78    var nameParts, nameSegment, mapValue, foundMap, lastIndex,
79        foundI, foundStarMap, startI, j, part, normalizedBaseParts,
80        baseParts = baseName && baseName.split("/"),
81        map = config.map,
82        starMap = (map && map["*"]) || {};
83
84    //Adjust any relative paths.
85    if (name) {
86      name = name.split('/');
87      lastIndex = name.length - 1;
88
89      // If wanting node ID compatibility, strip .js from end
90      // of IDs. Have to do this here, and not in nameToUrl
91      // because node allows either .js or non .js to map
92      // to same file.
93      if (config.nodeIdCompat && jsSuffixRegExp.test(name[lastIndex])) {
94        name[lastIndex] = name[lastIndex].replace(jsSuffixRegExp, "");
95      }
96
97      // Starts with a '.' so need the baseName
98      if (name[0].charAt(0) === '.' && baseParts) {
99        //Convert baseName to array, and lop off the last part,
100        //so that . matches that 'directory' and not name of the baseName's
101        //module. For instance, baseName of 'one/two/three', maps to
102        //'one/two/three.js', but we want the directory, 'one/two' for
103        //this normalization.
104        normalizedBaseParts = baseParts.slice(0, baseParts.length - 1);
105        name = normalizedBaseParts.concat(name);
106      }
107
108      //start trimDots
109      for (i = 0; i < name.length; i++) {
110        part = name[i];
111        if (part === '.') {
112          name.splice(i, 1);
113          i -= 1;
114        } else if (part === '..') {
115          // If at the start, or previous value is still ..,
116          // keep them so that when converted to a path it may
117          // still work when converted to a path, even though
118          // as an ID it is less than ideal. In larger point
119          // releases, may be better to just kick out an error.
120          if (i === 0 || (i === 1 && name[2] === '..') || name[i - 1] === '..') {
121            continue;
122          } else if (i > 0) {
123            name.splice(i - 1, 2);
124            i -= 2;
125          }
126        }
127      }
128      //end trimDots
129
130      name = name.join('/');
131    }
132
133    //Apply map config if available.
134    if ((baseParts || starMap) && map) {
135      nameParts = name.split('/');
136
137      for (i = nameParts.length; i > 0; i -= 1) {
138        nameSegment = nameParts.slice(0, i).join("/");
139
140        if (baseParts) {
141          //Find the longest baseName segment match in the config.
142          //So, do joins on the biggest to smallest lengths of baseParts.
143          for (j = baseParts.length; j > 0; j -= 1) {
144            mapValue = map[baseParts.slice(0, j).join("/")];
145
146            //baseName segment has config, find if it has one for
147            //this name.
148            if (mapValue) {
149              mapValue = mapValue[nameSegment];
150              if (mapValue) {
151                //Match, update name to the new value.
152                foundMap = mapValue;
153                foundI = i;
154                break;
155              }
156            }
157          }
158        }
159
160        if (foundMap) {
161          break;
162        }
163
164        //Check for a star map match, but just hold on to it,
165        //if there is a shorter segment match later in a matching
166        //config, then favor over this star map.
167        if (!foundStarMap && starMap && starMap[nameSegment]) {
168          foundStarMap = starMap[nameSegment];
169          startI = i;
170        }
171      }
172
173      if (!foundMap && foundStarMap) {
174        foundMap = foundStarMap;
175        foundI = startI;
176      }
177
178      if (foundMap) {
179        nameParts.splice(0, foundI, foundMap);
180        name = nameParts.join('/');
181      }
182    }
183
184    return name;
185  }

```

```

186
187 function makeRequire(relName, forceSync) {
188   return function () {
189     //A version of a require function that passes a moduleName
190     //value for items that may need to
191     //look up paths relative to the moduleName
192     var args = aps.call(arguments, 0);
193
194     //If first arg is not require('string'), and there is only
195     //one arg, it is the array form without a callback. Insert
196     //a null so that the following concat is correct.
197     if (typeof args[0] !== 'string' && args.length === 1) {
198       args.push(null);
199     }
200     return req.apply(undef, args.concat([relName, forceSync]));
201   };
202 }
203
204 function makeNormalize(relName) {
205   return function (name) {
206     return normalize(name, relName);
207   };
208 }
209
210 function makeLoad(depName) {
211   return function (value) {
212     defined[depName] = value;
213   };
214 }
215
216 function callDep(name) {
217   if (hasProp(waiting, name)) {
218     var args = waiting[name];
219     delete waiting[name];
220     defining[name] = true;
221     main.apply(undef, args);
222   }
223
224   if (!hasProp(defined, name) && !hasProp(defining, name)) {
225     throw new Error('No ' + name);
226   }
227   return defined[name];
228 }
229
230 //Turns a plugin!resource to [plugin, resource]
231 //with the plugin being undefined if the name
232 //did not have a plugin prefix.
233 function splitPrefix(name) {
234   var prefix,
235       index = name ? name.indexOf('!') : -1;
236   if (index > -1) {
237     prefix = name.substring(0, index);
238     name = name.substring(index + 1, name.length);
239   }
240   return [prefix, name];
241 }
242
243 //Creates a parts array for a relName where first part is plugin ID,
244 //second part is resource ID. Assumes relName has already been normalized.
245 function makeRelParts(relName) {
246   return relName ? splitPrefix(relName) : [];
247 }
248
249 /**
250  * Makes a name map, normalizing the name, and using a plugin
251  * for normalization if necessary. Grabs a ref to plugin
252  * too, as an optimization.
253  */
254 makeMap = function (name, relParts) {
255   var plugin,
256       parts = splitPrefix(name),
257       prefix = parts[0],
258       relResourceName = relParts[1];
259
260   name = parts[1];
261
262   if (prefix) {
263     prefix = normalize(prefix, relResourceName);
264     plugin = callDep(prefix);
265   }
266
267   //Normalize according
268   if (prefix) {
269     if (plugin && plugin.normalize) {
270       name = plugin.normalize(name, makeNormalize(relResourceName));
271     } else {
272       name = normalize(name, relResourceName);
273     }
274   } else {
275     name = normalize(name, relResourceName);
276     parts = splitPrefix(name);
277     prefix = parts[0];
278     name = parts[1];
279     if (prefix) {
280       plugin = callDep(prefix);
281     }
282   }
283
284   //Using ridiculous property names for space reasons
285   return {
286     f: prefix ? prefix + '!' + name : name, //fullName
287     n: name,
288     pr: prefix,
289     p: plugin
290   };
291 };
292
293 function makeConfig(name) {
294   return function () {
295     return [config && config.config && config.config[name]] || {};
296   };
297 }
298
299 handlers = {
300   require: function (name) {

```

```

301     return makeRequire(name);
302 },
303 exports: function (name) {
304     var e = defined(name);
305     if (typeof e !== 'undefined') {
306         return e;
307     } else {
308         return (defined[name] = {});
309     }
310 },
311 module: function (name) {
312     return {
313         id: name,
314         uri: "",
315         exports: defined(name),
316         config: makeConfig(name)
317     };
318 }
319 };
320
321 main = function (name, deps, callback, relName) {
322     var cjsModule, depName, ret, map, i, relParts;
323     args = [];
324     callbackType = typeof callback;
325     usingExports;
326
327     //Use name if no relName
328     relName = relName || name;
329     relParts = makeRelParts(relName);
330
331     //Call the callback to define the module, if necessary.
332     if (callbackType === 'undefined' || callbackType === 'function') {
333         //Pull out the defined dependencies and pass the ordered
334         //values to the callback.
335         //Default to [require, exports, module] if no deps
336         deps = (deps.length && callback.length ? ['require', 'exports', 'module'] : deps);
337         for (i = 0; i < deps.length; i += 1) {
338             map = makeMap(deps[i], relParts);
339             depName = map.f;
340
341             //Fast path CommonJS standard dependencies.
342             if (depName === "require") {
343                 args[i] = handlers.require(name);
344             } else if (depName === "exports") {
345                 //CommonJS module spec 1.1
346                 args[i] = handlers.exports(name);
347                 usingExports = true;
348             } else if (depName === "module") {
349                 //CommonJS module spec 1.1
350                 cjsModule = args[i] = handlers.module(name);
351             } else if (hasProp(defined, depName) ||
352                 hasProp(waiting, depName) ||
353                 hasProp(defining, depName)) {
354                 args[i] = callDep(depName);
355             } else if (map.p) {
356                 map.p.load(map.n, makeRequire(relName, true), makeLoad(depName), {});
357                 args[i] = defined(depName);
358             } else {
359                 throw new Error(name + ' missing ' + depName);
360             }
361         }
362
363         ret = callback ? callback.apply(defined[name], args) : undefined;
364
365         if (name) {
366             //If setting exports via "module" is in play,
367             //favor that over return value and exports. After that,
368             //favor a non-undefined return value over exports use.
369             if (cjsModule && cjsModule.exports !== undef &&
370                 cjsModule.exports !== defined[name]) {
371                 defined[name] = cjsModule.exports;
372             } else if (ret !== undef || !usingExports) {
373                 //Use the return value from the function.
374                 defined[name] = ret;
375             }
376         }
377     } else if (name) {
378         //May just be an object definition for the module. Only
379         //worry about defining if have a module name.
380         defined[name] = callback;
381     }
382 };
383
384 requirejs = require = req = function (deps, callback, relName, forceSync, alt) {
385     if (typeof deps === "string") {
386         if (handlers.deps) {
387             //callback in this case is really relName
388             return handlers.deps(callback);
389         }
390         //Just return the module wanted. In this scenario, the
391         //deps arg is the module name, and second arg (if passed)
392         //is just the relName.
393         //Normalize module name, if it contains . or ..
394         return callDep(makeMap(deps, makeRelParts(callback)).f);
395     } else if (!deps.splice) {
396         //deps is a config object, not an array.
397         config = deps;
398         if (config.deps) {
399             req(config.deps, config.callback);
400         }
401         if (!callback) {
402             return;
403         }
404
405         if (callback.splice) {
406             //callback is an array, which means it is a dependency list.
407             //Adjust args if there are dependencies
408             deps = callback;
409             callback = relName;
410             relName = null;
411         } else {
412             deps = undef;
413         }
414     }
415

```

```

416 //Support require(['a'])
417 callback = callback || function () {};
418
419 //If relName is a function, it is an errback handler,
420 //so remove it.
421 if (typeof relName === 'function') {
422     relName = forceSync;
423     forceSync = alt;
424 }
425
426 //Simulate async callback;
427 if (forceSync) {
428     main(undef, deps, callback, relName);
429 } else {
430     //Using a non-zero value because of concern for what old browsers
431     //do, and latest browsers "upgrade" to 4 if lower value is used:
432     //http://www.whatwg.org/specs/web-apps/current-work/multipage/timers.html#dom-windowtimers-settimeout;
433     //If want a value immediately, use require('id') instead -- something
434     //that works in almond on the global level, but not guaranteed and
435     //unlikely to work in other AMD implementations.
436     setTimeout(function () {
437         main(undef, deps, callback, relName);
438     }, 4);
439 }
440
441 return req;
442 };
443
444 /**
445  * Just drops the config on the floor, but returns req in case
446  * the config return value is used.
447  */
448 req.config = function (cfg) {
449     return req(cfg);
450 };
451
452 /**
453  * Expose module registry for debugging and tooling
454  */
455 requirejs._defined = defined;
456
457 define = function (name, deps, callback) {
458     if (typeof name !== 'string') {
459         throw new Error("See almond README: incorrect module build, no module name");
460     }
461
462     //This module may not have dependencies
463     if (!deps.splice) {
464         //deps is not an array, so probably means
465         //an object literal or factory function for
466         //the value. Adjust args.
467         callback = deps;
468         deps = [];
469     }
470
471     if (!('hasProp' in name) && 'hasProp' in waiting) {
472         waiting[name] = [name, deps, callback];
473     }
474 };
475
476 define.amd = {
477     jQuery: true
478 };
479 }());
480
481 S2.requirejs = requirejs; S2.require = require; S2.define = define;
482 }
483 }());
484 S2.define("almond", function() {});
485
486 /* global jQuery:false, $:false */
487 S2.define("jquery", [], function () {
488     var _$ = jQuery || $;
489
490     if (_$ === null && console && console.error) {
491         console.error(
492             'Select2: An instance of jQuery or a jQuery-compatible library was not ' +
493             'found. Make sure that you are including jQuery before Select2 on your ' +
494             'web page.'
495         );
496     }
497
498     return _$;
499 });
500
501 S2.define('select2/utils',[
502     'jquery'
503 ], function ($) {
504     var Utils = {};
505
506     Utils.Extend = function (ChildClass, SuperClass) {
507         var __hasProp = {}.hasOwnProperty;
508
509         function BaseConstructor () {
510             this.constructor = ChildClass;
511         }
512
513         for (var key in SuperClass) {
514             if (__hasProp.call(SuperClass, key)) {
515                 ChildClass[key] = SuperClass[key];
516             }
517         }
518
519         BaseConstructor.prototype = SuperClass.prototype;
520         ChildClass.prototype = new BaseConstructor();
521         ChildClass.__super__ = SuperClass.prototype;
522
523         return ChildClass;
524     };
525
526     function getMethods (theClass) {
527         var proto = theClass.prototype;
528
529         var methods = [];
530
531         for (var i = 0, l = methods.length; i < l; i++) {
532             // ...
533         }
534     }

```

```

531 for (var methodName in proto) {
532     var m = proto[methodName];
533
534     if (typeof m !== 'function') {
535         continue;
536     }
537
538     if (methodName === 'constructor') {
539         continue;
540     }
541
542     methods.push(methodName);
543 }
544
545 return methods;
546 }
547
548 Utils.Decorate = function (SuperClass, DecoratorClass) {
549     var decoratedMethods = getMethods(DecoratorClass);
550     var superMethods = getMethods(SuperClass);
551
552     function DecoratedClass () {
553         var unshift = Array.prototype.unshift;
554
555         var argCount = DecoratorClass.prototype.constructor.length;
556
557         var calledConstructor = SuperClass.prototype.constructor;
558
559         if (argCount > 0) {
560             unshift.call(arguments, SuperClass.prototype.constructor);
561
562             calledConstructor = DecoratorClass.prototype.constructor;
563         }
564
565         calledConstructor.apply(this, arguments);
566     }
567
568     DecoratedClass.displayName = SuperClass.displayName;
569
570     function ctr () {
571         this.constructor = DecoratedClass;
572     }
573
574     DecoratedClass.prototype = new ctr();
575
576     for (var m = 0; m < superMethods.length; m++) {
577         var superMethod = superMethods[m];
578
579         DecoratedClass.prototype[superMethod] =
580             SuperClass.prototype[superMethod];
581     }
582
583     var calledMethod = function (methodName) {
584         // Stub out the original method if it's not decorating an actual method
585         var originalMethod = function () {};
586
587         if (methodName in DecoratedClass.prototype) {
588             originalMethod = DecoratedClass.prototype[methodName];
589         }
590
591         var decoratedMethod = DecoratedClass.prototype[methodName];
592
593         return function () {
594             var unshift = Array.prototype.unshift;
595
596             unshift.call(arguments, originalMethod);
597
598             return decoratedMethod.apply(this, arguments);
599         };
600     };
601
602     for (var d = 0; d < decoratedMethods.length; d++) {
603         var decoratedMethod = decoratedMethods[d];
604
605         DecoratedClass.prototype[decoratedMethod] = calledMethod(decoratedMethod);
606     }
607
608     return DecoratedClass;
609 };
610
611 var Observable = function () {
612     this.listeners = {};
613 };
614
615 Observable.prototype.on = function (event, callback) {
616     this.listeners = this.listeners || {};
617
618     if (event in this.listeners) {
619         this.listeners[event].push(callback);
620     } else {
621         this.listeners[event] = [callback];
622     }
623 };
624
625 Observable.prototype.trigger = function (event) {
626     var slice = Array.prototype.slice;
627     var params = slice.call(arguments, 1);
628
629     this.listeners = this.listeners || {};
630
631     // Params should always come in as an array
632     if (params === null) {
633         params = [];
634     }
635
636     // If there are no arguments to the event, use a temporary object
637     if (params.length === 0) {
638         params.push({});
639     }
640
641     // Set the `_type` of the first object to the event
642     params[0]._type = event;
643
644     if (event in this.listeners) {
645         this.invoke(this.listeners[event], slice.call(arguments, 1));

```



```

646     }
647
648     if ('*' in this.listeners) {
649         this.invoke(this.listeners['*'], arguments);
650     }
651 };
652
653 Observable.prototype.invoke = function (listeners, params) {
654     for (var i = 0, len = listeners.length; i < len; i++) {
655         listeners[i].apply(this, params);
656     }
657 };
658
659 Utils.Observable = Observable;
660
661 Utils.generateChars = function (length) {
662     var chars = "";
663
664     for (var i = 0; i < length; i++) {
665         var randomChar = Math.floor(Math.random() * 36);
666         chars += randomChar.toString(36);
667     }
668
669     return chars;
670 };
671
672 Utils.bind = function (func, context) {
673     return function () {
674         func.apply(context, arguments);
675     };
676 };
677
678 Utils._convertData = function (data) {
679     for (var originalKey in data) {
680         var keys = originalKey.split('.');
681
682         var dataLevel = data;
683
684         if (keys.length === 1) {
685             continue;
686         }
687
688         for (var k = 0; k < keys.length; k++) {
689             var key = keys[k];
690
691             // Lowercase the first letter
692             // By default, dash-separated becomes camelCase
693             key = key.substring(0, 1).toLowerCase() + key.substring(1);
694
695             if (!(key in dataLevel)) {
696                 dataLevel[key] = {};
697             }
698
699             if (k === keys.length - 1) {
700                 dataLevel[key] = data[originalKey];
701             }
702
703             dataLevel = dataLevel[key];
704         }
705
706         delete data[originalKey];
707     }
708
709     return data;
710 };
711
712 Utils.hasScroll = function (index, el) {
713     // Adapted from the function created by @ShadowScripter
714     // and adapted by @BillBarry on the Stack Exchange Code Review website.
715     // The original code can be found at
716     // http://codereview.stackexchange.com/q/13338
717     // and was designed to be used with the Sizzle selector engine.
718
719     var $el = $(el);
720     var overflowX = el.style.overflowX;
721     var overflowY = el.style.overflowY;
722
723     //Check both x and y declarations
724     if (overflowX === overflowY &&
725         (overflowY === 'hidden' || overflowY === 'visible')) {
726         return false;
727     }
728
729     if (overflowX === 'scroll' || overflowY === 'scroll') {
730         return true;
731     }
732
733     return ($el.innerHeight() < el.scrollHeight ||
734         $el.innerWidth() < el.scrollWidth);
735 };
736
737 Utils.escapeMarkup = function (markup) {
738     var replaceMap = {
739         '\\': '&#92;',
740         '&': '&amp;',
741         '<': '&lt;',
742         '>': '&gt;',
743         '"': '&quot;',
744         "'"': '&#39;',
745         '/': '&#47;'
746     };
747
748     // Do not try to escape the markup if it's not a string
749     if (typeof markup !== 'string') {
750         return markup;
751     }
752
753     return String(markup).replace(/[&<>"'\/]/g, function (match) {
754         return replaceMap[match];
755     });
756 };
757
758 // Append an array of jQuery nodes to a given element.
759 Utils.appendMany = function ($element, $nodes) {
760     // jQuery 1.7.x does not support $.fn.append() with an array

```

```

761 // Fall back to a jQuery object collection using $.fn.add()
762 if ($ fn jquery substr(0, 3) === '1.7') {
763     var $jqNodes = $();
764
765     $ map $nodes function (node) {
766         $jqNodes = $jqNodes add node;
767     };
768
769     $nodes = $jqNodes;
770 }
771
772 $element append $nodes;
773 };
774
775 // Cache objects in Utils.__cache instead of $.data (see #4346)
776 Utils.__cache = {};
777
778 var id = 0;
779 Utils.GetUniqueElementId = function (element) {
780     // Get a unique element id. If element has no id,
781     // creates a new unique number, stores it in the id
782     // attribute and returns the new id.
783     // If an id already exists, it simply returns it.
784
785     var select2Id = element.getAttribute('data-select2-id');
786     if (select2Id == null) {
787         // If element has id, use it.
788         if (element.id) {
789             select2Id = element.id;
790             element.setAttribute('data-select2-id', select2Id);
791         } else {
792             element.setAttribute('data-select2-id', ++id);
793             select2Id = id.toString();
794         }
795     }
796     return select2Id;
797 };
798
799 Utils.StoreData = function (element, name, value) {
800     // Stores an item in the cache for a specified element.
801     // name is the cache key.
802     var id = Utils.GetUniqueElementId(element);
803     if (!Utils.__cache[id]) {
804         Utils.__cache[id] = {};
805     }
806
807     Utils.__cache[id][name] = value;
808 };
809
810 Utils.GetData = function (element, name) {
811     // Retrieves a value from the cache by its key (name)
812     // name is optional. If no name specified, return
813     // all cache items for the specified element.
814     // and for a specified element.
815     var id = Utils.GetUniqueElementId(element);
816     if (name) {
817         if (Utils.__cache[id]) {
818             if (Utils.__cache[id][name] != null) {
819                 return Utils.__cache[id][name];
820             }
821             return $(element).data(name); // Fallback to HTML5 data attribs.
822         }
823         return $(element).data(name); // Fallback to HTML5 data attribs.
824     } else {
825         return Utils.__cache[id];
826     }
827 };
828
829 Utils.RemoveData = function (element) {
830     // Removes all cached items for a specified element.
831     var id = Utils.GetUniqueElementId(element);
832     if (Utils.__cache[id] != null) {
833         delete Utils.__cache[id];
834     }
835
836     element.removeAttribute('data-select2-id');
837 };
838
839 return Utils;
840 };
841
842 S2.define('select2/results', [
843     'jquery',
844     './utils'
845 ], function ($, Utils) {
846     function Results ($element, options, dataAdapter) {
847         this.$element = $element;
848         this.data = dataAdapter;
849         this.options = options;
850
851         Results.__super__.constructor.call(this);
852     }
853
854     Utils.Extend(Results, Utils.Observable);
855
856     Results.prototype.render = function () {
857         var $results = $(
858             '<ul class="select2-results__options" role="listbox"></ul>'
859         );
860
861         if (this.options.get('multiple')) {
862             $results.attr('aria-multiselectable', 'true');
863         }
864
865         this.$results = $results;
866
867         return $results;
868     };
869
870     Results.prototype.clear = function () {
871         this.$results.empty();
872     };
873
874     Results.prototype.displayMessage = function (params) {
875         var escapeMarkup = this.options.get('escapeMarkup');

```

```

876
877 this.clear();
878 this.hideLoading();
879
880 var $message = $
881   '<li role="alert" aria-live="assertive" +
882   ' class="select2-results__option"></li>'
883 );
884
885 var message = this.options.get('translations').get(params.message);
886
887 $message.append(
888   escapeMarkup(
889     message.params.args
890   )
891 );
892
893 $message[0].className += ' select2-results__message';
894
895 this.$results.append($message);
896 };
897
898 Results.prototype.hideMessages = function () {
899   this.$results.find('.select2-results__message').remove();
900 };
901
902 Results.prototype.append = function (data) {
903   this.hideLoading();
904
905   var $options = [];
906
907   if (data.results == null || data.results.length === 0) {
908     if (this.$results.children().length === 0) {
909       this.trigger('results:message', {
910         message: 'noResults'
911       });
912     }
913
914     return;
915   }
916
917   data.results = this.sort(data.results);
918
919   for (var d = 0; d < data.results.length; d++) {
920     var item = data.results[d];
921
922     var $option = this.option(item);
923
924     $options.push($option);
925   }
926
927   this.$results.append($options);
928 };
929
930 Results.prototype.position = function ($results, $dropdown) {
931   var $resultsContainer = $dropdown.find('.select2-results');
932   $resultsContainer.append($results);
933 };
934
935 Results.prototype.sort = function (data) {
936   var sorter = this.options.get('sorter');
937
938   return sorter(data);
939 };
940
941 Results.prototype.highlightFirstItem = function () {
942   var $options = this.$results
943     .find('.select2-results__option[aria-selected]');
944
945   var $selected = $options.filter('[aria-selected=true]');
946
947   // Check if there are any selected options
948   if ($selected.length > 0) {
949     // If there are selected options, highlight the first
950     $selected.first().trigger('mouseenter');
951   } else {
952     // If there are no selected options, highlight the first option
953     // in the dropdown
954     $options.first().trigger('mouseenter');
955   }
956
957   this.ensureHighlightVisible();
958 };
959
960 Results.prototype.setClasses = function () {
961   var self = this;
962
963   this.data.current(function (selected) {
964     var selectedIds = $.map(selected, function (s) {
965       return s.id.toString();
966     });
967
968     var $options = self.$results
969       .find('.select2-results__option[aria-selected]');
970
971     $options.each(function () {
972       var $option = $(this);
973
974       var item = Utils.GetData(this, 'data');
975
976       // id needs to be converted to a string when comparing
977       var id = "" + item.id;
978
979       if ((item.element != null && item.element.selected) ||
980         (item.element == null && $inArray(id, selectedIds) > -1)) {
981         $option.attr('aria-selected', 'true');
982       } else {
983         $option.attr('aria-selected', 'false');
984       }
985     });
986
987   });
988 };
989
990 Results.prototype.showLoading = function (params) {

```

```

991     this.hideLoading();
992
993     var loadingMore = this.options.get('translations').get('searching');
994
995     var loading = {
996         disabled: true,
997         loading: true,
998         text: loadingMore.params()
999     };
1000     var $loading = this.option.loading;
1001     $loading.className += ' loading-results';
1002
1003     this.$results.prepend($loading);
1004 };
1005
1006 Results.prototype.hideLoading = function () {
1007     this.$results.find('.loading-results').remove();
1008 };
1009
1010 Results.prototype.option = function (data) {
1011     var option = document.createElement('li');
1012     option.className = 'select2-results__option';
1013
1014     var attrs = {
1015         'role': 'option',
1016         'aria-selected': 'false'
1017     };
1018
1019     var matches = window.Element.prototype.matches ||
1020         window.Element.prototype.msMatchesSelector ||
1021         window.Element.prototype.webkitMatchesSelector;
1022
1023     if ((data.element != null && matches.call(data.element, ':disabled')) ||
1024         (data.element == null && data.disabled)) {
1025         delete attrs['aria-selected'];
1026         attrs['aria-disabled'] = 'true';
1027     }
1028
1029     if (data.id == null) {
1030         delete attrs['aria-selected'];
1031     }
1032
1033     if (data._resultId != null) {
1034         option.id = data._resultId;
1035     }
1036
1037     if (data.title) {
1038         option.title = data.title;
1039     }
1040
1041     if (data.children) {
1042         attrs.role = 'group';
1043         attrs['aria-label'] = data.text;
1044         delete attrs['aria-selected'];
1045     }
1046
1047     for (var attr in attrs) {
1048         var val = attrs[attr];
1049
1050         option.setAttribute(attr, val);
1051     }
1052
1053     if (data.children) {
1054         var $option = $(option);
1055
1056         var label = document.createElement('strong');
1057         label.className = 'select2-results__group';
1058
1059         var $label = $(label);
1060         this.template(data, label);
1061
1062         var $children = [];
1063
1064         for (var c = 0, c < data.children.length; c++) {
1065             var child = data.children[c];
1066
1067             var $child = this.option(child);
1068
1069             $children.push($child);
1070         }
1071
1072         var $childrenContainer = $('<ul></ul>', {
1073             'class': 'select2-results__options select2-results__options--nested'
1074         });
1075
1076         $childrenContainer.append($children);
1077
1078         $option.append(label);
1079         $option.append($childrenContainer);
1080     } else {
1081         this.template(data, option);
1082     }
1083
1084     Utils.StoreData(option, 'data', data);
1085
1086     return option;
1087 };
1088
1089 Results.prototype.bind = function (container, $container) {
1090     var self = this;
1091
1092     var id = container.id + '-results';
1093
1094     this.$results.attr('id', id);
1095
1096     container.on('results:all', function (params) {
1097         self.clear();
1098         self.append(params.data);
1099
1100         if (container.isOpen()) {
1101             self.setClasses();
1102             self.highlightFirstItem();
1103         }
1104     });
1105

```

```

1106 container.on('results:append', function (params) {
1107     self.append(params.data);
1108
1109     if (container.isOpen()) {
1110         self.setClasses();
1111     }
1112 });
1113
1114 container.on('query', function (params) {
1115     self.hideMessages();
1116     self.showLoading(params);
1117 });
1118
1119 container.on('select', function () {
1120     if (!container.isOpen()) {
1121         return;
1122     }
1123
1124     self.setClasses();
1125
1126     if (self.options.get('scrollAfterSelect')) {
1127         self.highlightFirstItem();
1128     }
1129 });
1130
1131 container.on('unselect', function () {
1132     if (!container.isOpen()) {
1133         return;
1134     }
1135
1136     self.setClasses();
1137
1138     if (self.options.get('scrollAfterSelect')) {
1139         self.highlightFirstItem();
1140     }
1141 });
1142
1143 container.on('open', function () {
1144     // When the dropdown is open, aria-expanded="true"
1145     self.$results.attr('aria-expanded', 'true');
1146     self.$results.attr('aria-hidden', 'false');
1147
1148     self.setClasses();
1149     self.ensureHighlightVisible();
1150 });
1151
1152 container.on('close', function () {
1153     // When the dropdown is closed, aria-expanded="false"
1154     self.$results.attr('aria-expanded', 'false');
1155     self.$results.attr('aria-hidden', 'true');
1156     self.$results.removeAttr('aria-activedescendant');
1157 });
1158
1159 container.on('results:toggle', function () {
1160     var $highlighted = self.getHighlightedResults();
1161
1162     if ($highlighted.length === 0) {
1163         return;
1164     }
1165
1166     $highlighted.trigger('mouseup');
1167 });
1168
1169 container.on('results:select', function () {
1170     var $highlighted = self.getHighlightedResults();
1171
1172     if ($highlighted.length === 0) {
1173         return;
1174     }
1175
1176     var data = Utils.GetData($highlighted[0], 'data');
1177
1178     if ($highlighted.attr('aria-selected') === 'true') {
1179         self.trigger('close', {});
1180     } else {
1181         self.trigger('select', {
1182             data: data
1183         });
1184     }
1185 });
1186
1187 container.on('results:previous', function () {
1188     var $highlighted = self.getHighlightedResults();
1189
1190     var $options = self.$results.find('[aria-selected]');
1191
1192     var currentIndex = $options.index($highlighted);
1193
1194     // If we are already at the top, don't move further
1195     // If no options, currentIndex will be -1
1196     if (currentIndex <= 0) {
1197         return;
1198     }
1199
1200     var nextIndex = currentIndex - 1;
1201
1202     // If none are highlighted, highlight the first
1203     if ($highlighted.length === 0) {
1204         nextIndex = 0;
1205     }
1206
1207     var $next = $options.eq(nextIndex);
1208
1209     $next.trigger('mouseenter');
1210
1211     var currentOffset = self.$results.offset().top;
1212     var nextTop = $next.offset().top;
1213     var nextOffset = self.$results.scrollTop() + (nextTop - currentOffset);
1214
1215     if (nextIndex === 0) {
1216         self.$results.scrollTop(0);
1217     } else if (nextTop - currentOffset < 0) {
1218         self.$results.scrollTop(nextOffset);
1219     }
1220 });

```

```

1221
1222 container.on('results:next', function () {
1223     var $highlighted = self.getHighlightedResults();
1224
1225     var $options = self.$results.find('[aria-selected]');
1226
1227     var currentIndex = $options.index($highlighted);
1228
1229     var nextIndex = currentIndex + 1;
1230
1231     // If we are at the last option, stay there
1232     if (nextIndex >= $options.length) {
1233         return;
1234     }
1235
1236     var $next = $options.eq(nextIndex);
1237
1238     $next.trigger('mouseenter');
1239
1240     var currentOffset = self.$results.offset().top +
1241         self.$results.outerHeight(false);
1242     var nextBottom = $next.offset().top + $next.outerHeight(false);
1243     var nextOffset = self.$results.scrollTop() + nextBottom - currentOffset;
1244
1245     if (nextIndex === 0) {
1246         self.$results.scrollTop(0);
1247     } else if (nextBottom > currentOffset) {
1248         self.$results.scrollTop(nextOffset);
1249     }
1250 });
1251
1252 container.on('results:focus', function (params) {
1253     params.element.addClass('select2-results__option--highlighted');
1254 });
1255
1256 container.on('results:message', function (params) {
1257     self.displayMessage(params);
1258 });
1259
1260 if ($fn.mousewheel) {
1261     this.$results.on('mousewheel', function (e) {
1262         var top = self.$results.scrollTop();
1263
1264         var bottom = self.$results.get(0).scrollHeight - top + e.deltaY;
1265
1266         var isAtTop = e.deltaY > 0 && top - e.deltaY <= 0;
1267         var isAtBottom = e.deltaY < 0 && bottom <= self.$results.height();
1268
1269         if (isAtTop) {
1270             self.$results.scrollTop(0);
1271
1272             e.preventDefault();
1273             e.stopPropagation();
1274         } else if (isAtBottom) {
1275             self.$results.scrollTop(
1276                 self.$results.get(0).scrollHeight - self.$results.height()
1277             );
1278
1279             e.preventDefault();
1280             e.stopPropagation();
1281         }
1282     });
1283 }
1284
1285 this.$results.on('mouseup', '.select2-results__option[aria-selected]',
1286     function (evt) {
1287         var $this = $(this);
1288
1289         var data = Utils.GetData(this, 'data');
1290
1291         if ($this.attr('aria-selected') === 'true') {
1292             if (self.options.get('multiple')) {
1293                 self.trigger('unselect', {
1294                     originalEvent: evt,
1295                     data: data
1296                 });
1297             } else {
1298                 self.trigger('close', {});
1299             }
1300
1301             return;
1302         }
1303
1304         self.trigger('select', {
1305             originalEvent: evt,
1306             data: data
1307         });
1308     });
1309
1310 this.$results.on('mouseenter', '.select2-results__option[aria-selected]',
1311     function (evt) {
1312         var data = Utils.GetData(this, 'data');
1313
1314         self.getHighlightedResults()
1315             .removeClass('select2-results__option--highlighted');
1316
1317         self.trigger('results:focus', {
1318             data: data,
1319             element: $this
1320         });
1321     });
1322 };
1323
1324 Results.prototype.getHighlightedResults = function () {
1325     var $highlighted = this.$results
1326         .find('.select2-results__option--highlighted');
1327
1328     return $highlighted;
1329 };
1330
1331 Results.prototype.destroy = function () {
1332     this.$results.remove();
1333 };
1334
1335 Results.prototype.ensureHighlightVisible = function () {

```

```

1336 var $highlighted = this.getHighlightedResults();
1337
1338 if ($highlighted.length === 0) {
1339     return;
1340 }
1341
1342 var $options = this.$results.find('[aria-selected]');
1343
1344 var currentIndex = $options.index($highlighted);
1345
1346 var currentOffset = this.$results.offset().top;
1347 var nextTop = $highlighted.offset().top;
1348 var nextOffset = this.$results.scrollTop() + (nextTop - currentOffset);
1349
1350 var offsetDelta = nextTop - currentOffset;
1351 nextOffset -= $highlighted.outerHeight(false) * 2;
1352
1353 if (currentIndex <= 2) {
1354     this.$results.scrollTop(0);
1355 } else if (offsetDelta > this.$results.outerHeight() || offsetDelta < 0) {
1356     this.$results.scrollTop(nextOffset);
1357 }
1358 };
1359
1360 Results.prototype template = function (result, container) {
1361     var template = this.options.get('templateResult');
1362     var escapeMarkup = this.options.get('escapeMarkup');
1363
1364     var content = template(result, container);
1365
1366     if (content == null) {
1367         container.style.display = 'none';
1368     } else if (typeof content === 'string') {
1369         container.innerHTML = escapeMarkup(content);
1370     } else {
1371         $(container).append(content);
1372     }
1373 };
1374
1375 return Results;
1376 });
1377
1378 S2.define('select2/keys',[
1379     function () {
1380         var KEYS = {
1381             BACKSPACE: 8,
1382             TAB: 9,
1383             ENTER: 13,
1384             SHIFT: 16,
1385             CTRL: 17,
1386             ALT: 18,
1387             ESC: 27,
1388             SPACE: 32,
1389             PAGE_UP: 33,
1390             PAGE_DOWN: 34,
1391             END: 35,
1392             HOME: 36,
1393             LEFT: 37,
1394             UP: 38,
1395             RIGHT: 39,
1396             DOWN: 40,
1397             DELETE: 46
1398         };
1399     },
1400     function KEYS
1401 ]);
1402
1403 S2.define('select2/selection/base',[
1404     'jquery',
1405     './utils',
1406     './keys',
1407     function ($, Utils, KEYS) {
1408         function BaseSelection ($element, options) {
1409             this.$element = $element;
1410             this.options = options;
1411
1412             BaseSelection.__super__.constructor.call(this);
1413         }
1414
1415         Utils.Extend(BaseSelection, Utils.Observable);
1416
1417         BaseSelection.prototype.render = function () {
1418             var $selection = $(
1419                 '<span class="select2-selection" role="combobox" ' +
1420                 'aria-haspopup="true" aria-expanded="false">' +
1421                 '</span>'
1422             );
1423
1424             this._tabindex = 0;
1425
1426             if (Utils.GetData(this.$element[0], 'old-tabindex') != null) {
1427                 this._tabindex = Utils.GetData(this.$element[0], 'old-tabindex');
1428             } else if (this.$element.attr('tabindex') != null) {
1429                 this._tabindex = this.$element.attr('tabindex');
1430             }
1431
1432             $selection.attr('title', this.$element.attr('title'));
1433             $selection.attr('tabindex', this._tabindex);
1434             $selection.attr('aria-disabled', 'false');
1435
1436             this.$selection = $selection;
1437
1438             return $selection;
1439         };
1440
1441         BaseSelection.prototype.bind = function (container, $container) {
1442             var self = this;
1443
1444             var resultsId = container.id + '-results';
1445
1446             this.container = container;
1447
1448             this.$selection.on('focus', function (evt) {
1449                 self.trigger('focus', evt);
1450             });

```

```

1451 });
1452
1453 this.$selection.on('blur', function (evt) {
1454     self._handleBlur(evt);
1455 });
1456
1457 this.$selection.on('keydown', function (evt) {
1458     self.trigger('keypress', evt);
1459
1460     if (evt.which === KEYS.SPACE) {
1461         evt.preventDefault();
1462     }
1463 });
1464
1465 container.on('results:focus', function (params) {
1466     self.$selection.attr('aria-activedescendant', params.data._resultId);
1467 });
1468
1469 container.on('selection:update', function (params) {
1470     self.update(params.data);
1471 });
1472
1473 container.on('open', function () {
1474     // When the dropdown is open, aria-expanded="true"
1475     self.$selection.attr('aria-expanded', 'true');
1476     self.$selection.attr('aria-owns', resultsId);
1477
1478     self._attachCloseHandler(container);
1479 });
1480
1481 container.on('close', function () {
1482     // When the dropdown is closed, aria-expanded="false"
1483     self.$selection.attr('aria-expanded', 'false');
1484     self.$selection.removeAttr('aria-activedescendant');
1485     self.$selection.removeAttr('aria-owns');
1486
1487     self.$selection.trigger('focus');
1488
1489     self._detachCloseHandler(container);
1490 });
1491
1492 container.on('enable', function () {
1493     self.$selection.attr('tabindex', self._tabindex);
1494     self.$selection.attr('aria-disabled', 'false');
1495 });
1496
1497 container.on('disable', function () {
1498     self.$selection.attr('tabindex', '-1');
1499     self.$selection.attr('aria-disabled', 'true');
1500 });
1501 };
1502
1503 BaseSelection.prototype._handleBlur = function (evt) {
1504     var self = this;
1505
1506     // This needs to be delayed as the active element is the body when the tab
1507     // key is pressed, possibly along with others.
1508     window.setTimeout(function () {
1509         // Don't trigger `blur` if the focus is still in the selection
1510         if (
1511             document.activeElement === self.$selection[0] ||
1512             ($contains(self.$selection[0], document.activeElement))
1513         ) {
1514             return;
1515         }
1516
1517         self.trigger('blur', evt);
1518     }, 1);
1519 };
1520
1521 BaseSelection.prototype._attachCloseHandler = function (container) {
1522
1523     $(document.body).on('mousedown.select2.' + container.id, function (e) {
1524         var $target = $(e.target);
1525
1526         var $select = $target.closest('.select2');
1527
1528         var $all = $select.select2('container--open');
1529
1530         $all.each(function () {
1531             if (this !== $select[0]) {
1532                 return;
1533             }
1534
1535             var $element = Utils.GetData(this, 'element');
1536
1537             $element.select2('close');
1538         });
1539     });
1540 };
1541
1542 BaseSelection.prototype._detachCloseHandler = function (container) {
1543     $(document.body).off('mousedown.select2.' + container.id);
1544 };
1545
1546 BaseSelection.prototype.position = function ($selection, $container) {
1547     var $selectionContainer = $container.find('.selection');
1548     $selectionContainer.append($selection);
1549 };
1550
1551 BaseSelection.prototype.destroy = function () {
1552     this._detachCloseHandler(this.container);
1553 };
1554
1555 BaseSelection.prototype.update = function (data) {
1556     throw new Error('The `update` method must be defined in child classes.');
```



```

1566 BaseSelection.prototype.isEnabled = function () {
1567     return !this.isDisabled();
1568 };
1569
1570 /**
1571  * Helper method to abstract the "disabled" state of this object.
1572  *
1573  * @return {true} if the disabled option is true.
1574  * @return {false} if the disabled option is false.
1575  */
1576 BaseSelection.prototype.isDisabled = function () {
1577     return this.options.get('disabled');
1578 };
1579
1580 return BaseSelection;
1581 });
1582
1583 S2.define('select2/selection/single',[
1584     'jquery',
1585     './base',
1586     './utils',
1587     './keys'
1588 ], function ($, BaseSelection, Utils, KEYS) {
1589     function SingleSelection () {
1590         SingleSelection.__super__.constructor.apply(this, arguments);
1591     }
1592
1593     Utils.Extend(SingleSelection, BaseSelection);
1594
1595     SingleSelection.prototype.render = function () {
1596         var $selection = SingleSelection.__super__.render.call(this);
1597
1598         $selection.addClass('select2-selection--single');
1599
1600         $selection.html(
1601             '<span class="select2-selection__rendered"></span>' +
1602             '<span class="select2-selection__arrow" role="presentation">' +
1603             '<b role="presentation"></b>' +
1604             '</span>'
1605         );
1606
1607         return $selection;
1608     };
1609
1610     SingleSelection.prototype.bind = function (container, $container) {
1611         var self = this;
1612
1613         SingleSelection.__super__.bind.apply(this, arguments);
1614
1615         var id = container.id + '-container';
1616
1617         this.$selection.find('.select2-selection__rendered')
1618             .attr('id', id)
1619             .attr('role', 'textbox')
1620             .attr('aria-readonly', 'true');
1621         this.$selection.attr('aria-labelledby', id);
1622
1623         this.$selection.on('mousedown', function (evt) {
1624             // Only respond to left clicks
1625             if (evt.which !== 1) {
1626                 return;
1627             }
1628
1629             self.trigger('toggle', {
1630                 originalEvent: evt
1631             });
1632         });
1633
1634         this.$selection.on('focus', function (evt) {
1635             // User focuses on the container
1636         });
1637
1638         this.$selection.on('blur', function (evt) {
1639             // User exits the container
1640         });
1641
1642         container.on('focus', function (evt) {
1643             if (!container.isOpen()) {
1644                 self.$selection.trigger('focus');
1645             }
1646         });
1647     };
1648
1649     SingleSelection.prototype.clear = function () {
1650         var $rendered = this.$selection.find('.select2-selection__rendered');
1651         $rendered.empty();
1652         $rendered.removeAttr('title'); // clear tooltip on empty
1653     };
1654
1655     SingleSelection.prototype.display = function (data, container) {
1656         var template = this.options.get('templateSelection');
1657         var escapeMarkup = this.options.get('escapeMarkup');
1658
1659         return escapeMarkup(template(data, container));
1660     };
1661
1662     SingleSelection.prototype.selectionContainer = function () {
1663         return $('<span></span>');
1664     };
1665
1666     SingleSelection.prototype.update = function (data) {
1667         if (data.length === 0) {
1668             this.clear();
1669             return;
1670         }
1671
1672         var selection = data[0];
1673
1674         var $rendered = this.$selection.find('.select2-selection__rendered');
1675         var formatted = this.display(selection, $rendered);
1676
1677         $rendered.empty().append(formatted);
1678
1679         var title = selection.title || selection.text;
1680

```

```

1681     if (title) {
1682         $rendered.attr('title', title);
1683     } else {
1684         $rendered.removeAttr('title');
1685     }
1686 };
1687
1688 return SingleSelection;
1689 });
1690
1691 S2.define('select2/selection/multiple',[
1692     'jquery',
1693     './base',
1694     './utils'
1695 ], function ($, BaseSelection, Utils) {
1696     function MultipleSelection ($element, options) {
1697         MultipleSelection.__super__.constructor.apply(this, arguments);
1698     }
1699
1700     Utils.Extend(MultipleSelection, BaseSelection);
1701
1702     MultipleSelection.prototype.render = function () {
1703         var $selection = MultipleSelection.__super__.render.call(this);
1704
1705         $selection.addClass('select2-selection--multiple');
1706
1707         $selection.html(
1708             '<ul class="select2-selection__rendered"></ul>'
1709         );
1710
1711         return $selection;
1712     };
1713
1714     MultipleSelection.prototype.bind = function (container, $container) {
1715         var self = this;
1716
1717         MultipleSelection.__super__.bind.apply(this, arguments);
1718
1719         this.$selection.on('click', function (evt) {
1720             self.trigger('toggle', {
1721                 originalEvent: evt
1722             });
1723         });
1724
1725         this.$selection.on(
1726             'click',
1727             '.select2-selection__choice__remove',
1728             function (evt) {
1729                 // Ignore the event if it is disabled
1730                 if (self.isDisabled()) {
1731                     return;
1732                 }
1733
1734                 var $remove = $(this);
1735                 var $selection = $remove.parent();
1736
1737                 var data = Utils.GetData($selection[0], 'data');
1738
1739                 self.trigger('unselect', {
1740                     originalEvent: evt,
1741                     data: data
1742                 });
1743             }
1744         );
1745     };
1746
1747     MultipleSelection.prototype.clear = function () {
1748         var $rendered = this.$selection.find('.select2-selection__rendered');
1749         $rendered.empty();
1750         $rendered.removeAttr('title');
1751     };
1752
1753     MultipleSelection.prototype.display = function (data, container) {
1754         var template = this.options.get('templateSelection');
1755         var escapeMarkup = this.options.get('escapeMarkup');
1756
1757         return escapeMarkup(template(data, container));
1758     };
1759
1760     MultipleSelection.prototype.selectionContainer = function () {
1761         var $container = $(
1762             '<li class="select2-selection__choice">' +
1763                 '<span class="select2-selection__choice__remove" role="presentation">' +
1764                     '&times;' +
1765                 '</span>' +
1766             '</li>'
1767         );
1768
1769         return $container;
1770     };
1771
1772     MultipleSelection.prototype.update = function (data) {
1773         this.clear();
1774
1775         if (data.length === 0) {
1776             return;
1777         }
1778
1779         var $selections = [];
1780
1781         for (var d = 0; d < data.length; d++) {
1782             var selection = data[d];
1783
1784             var $selection = this.selectionContainer();
1785             var formatted = this.display(selection, $selection);
1786
1787             $selection.append(formatted);
1788
1789             var title = selection.title || selection.text;
1790
1791             if (title) {
1792                 $selection.attr('title', title);
1793             }
1794
1795             Utils.StoreData($selection[0], 'data', selection);

```

```

1796     $selections.push($selection);
1797 }
1798 }
1799
1800 var $rendered = this.$selection.find('.select2-selection__rendered');
1801
1802 Utils.appendMany($rendered, $selections);
1803 },
1804
1805 return MultipleSelection;
1806 });
1807
1808 S2.define('select2/selection/placeholder',[
1809   './utils'
1810 ], function (Utils) {
1811   function Placeholder (decorated, $element, options) {
1812     this.placeholder = this.normalizePlaceholder(options.get('placeholder'));
1813
1814     decorated.call(this, $element, options);
1815   }
1816
1817   Placeholder.prototype.normalizePlaceholder = function (_ placeholder) {
1818     if (typeof placeholder === 'string') {
1819       placeholder = {
1820         id: "",
1821         text: placeholder
1822       };
1823     }
1824
1825     return placeholder;
1826   };
1827
1828   Placeholder.prototype.createPlaceholder = function (decorated, placeholder) {
1829     var $placeholder = this.selectionContainer();
1830
1831     $placeholder.html(this.display(placeholder));
1832     $placeholder.addClass('select2-selection__placeholder')
1833       .removeClass('select2-selection__choice');
1834
1835     return $placeholder;
1836   };
1837
1838   Placeholder.prototype.update = function (decorated, data) {
1839     var singlePlaceholder = {
1840       data.length == 1 && data[0].id != this.placeholder.id
1841     };
1842
1843     var multipleSelections = data.length > 1;
1844
1845     if (multipleSelections || singlePlaceholder) {
1846       return decorated.call(this, data);
1847     }
1848
1849     this.clear();
1850
1851     var $placeholder = this.createPlaceholder(this.placeholder);
1852
1853     this.$selection.find('.select2-selection__rendered').append($placeholder);
1854   };
1855
1856   return Placeholder;
1857 });
1858
1859 S2.define('select2/selection/allowClear',[
1860   'jquery',
1861   './keys',
1862   './utils'
1863 ], function ($, KEYS, Utils) {
1864   function AllowClear () { }
1865
1866   AllowClear.prototype.bind = function (decorated, container, $container) {
1867     var self = this;
1868
1869     decorated.call(this, container, $container);
1870
1871     if (this.placeholder == null) {
1872       if (this.options.get('debug') && window.console && console.error) {
1873         console.error(
1874           'Select2: The `allowClear` option should be used in combination ' +
1875           'with the `placeholder` option.'
1876         );
1877       }
1878     }
1879
1880     this.$selection.on('mousedown', '.select2-selection__clear',
1881       function (evt) {
1882         self._handleClear(evt);
1883       });
1884
1885     container.on('keypress', function (evt) {
1886       self._handleKeyboardClear(evt, container);
1887     });
1888
1889     AllowClear.prototype._handleClear = function (_ evt) {
1890       // Ignore the event if it is disabled
1891       if (this.isDisabled()) {
1892         return;
1893       }
1894
1895       var $clear = this.$selection.find('.select2-selection__clear');
1896
1897       // Ignore the event if nothing has been selected
1898       if ($clear.length === 0) {
1899         return;
1900       }
1901
1902       evt.stopPropagation();
1903
1904       var data = Utils.GetData($clear[0], 'data');
1905
1906       var previousVal = this.$element.val();
1907       this.$element.val(this.placeholder.id);
1908
1909       var unselectData = {
1910         data: data

```

```

1911 };
1912 this.trigger('clear', unselectData);
1913 if (unselectData prevented) {
1914     this.$element.val(previousVal);
1915     return;
1916 }
1917
1918 for (var d = 0; d < data.length; d++) {
1919     unselectData = {
1920         data: data[d]
1921     };
1922
1923     // Trigger the `unselect` event, so people can prevent it from being
1924     // cleared.
1925     this.trigger('unselect', unselectData);
1926
1927     // If the event was prevented, don't clear it out.
1928     if (unselectData prevented) {
1929         this.$element.val(previousVal);
1930         return;
1931     }
1932 }
1933
1934 this.$element.trigger('input').trigger('change');
1935
1936 this.trigger('toggle', {});
1937 };
1938
1939 AllowClear.prototype._handleKeyboardClear = function (_ evt, container) {
1940     if (container.isOpen()) {
1941         return;
1942     }
1943
1944     if (evt.which == KEYS.DELETE || evt.which == KEYS.BACKSPACE) {
1945         this._handleClear(evt);
1946     }
1947 };
1948
1949 AllowClear.prototype.update = function (decorated, data) {
1950     decorated.call(this, data);
1951
1952     if (this.$selection.find('select2-selection__placeholder').length > 0 ||
1953         data.length === 0) {
1954         return;
1955     }
1956
1957     var removeAll = this.options.get('translations').get('removeAllItems');
1958
1959     var $remove = $(
1960         '<span class="select2-selection__clear" title="' + removeAll() + '">' +
1961         '&times;' +
1962         '</span>'
1963     );
1964     Utils.StoreData($remove[0], 'data', data);
1965
1966     this.$selection.find('select2-selection__rendered').prepend($remove);
1967 };
1968
1969 return AllowClear;
1970 });
1971
1972 S2.define('select2/selection/search', [
1973     'jquery',
1974     './utils',
1975     './keys'
1976 ], function ($, Utils, KEYS) {
1977     function Search(decorated, $element, options) {
1978         decorated.call(this, $element, options);
1979     }
1980
1981     Search.prototype.render = function (decorated) {
1982         var $search = $(
1983             '<li class="select2-search select2-search--inline">' +
1984             '<input class="select2-search__field" type="search" tabindex="-1" +
1985             ' autocomplete="off" autocorrect="off" autocapitalize="none" +
1986             ' spellcheck="false" role="searchbox" aria-autocomplete="list" />' +
1987             '</li>'
1988         );
1989
1990         this.$searchContainer = $search;
1991         this.$search = $search.find('input');
1992
1993         var $rendered = decorated.call(this);
1994
1995         this._transferTabIndex();
1996
1997         return $rendered;
1998     };
1999
2000     Search.prototype.bind = function (decorated, container, $container) {
2001         var self = this;
2002
2003         var resultsId = container.id + '-results';
2004
2005         decorated.call(this, container, $container);
2006
2007         container.on('open', function () {
2008             self.$search.attr('aria-controls', resultsId);
2009             self.$search.trigger('focus');
2010         });
2011
2012         container.on('close', function () {
2013             self.$search.val('');
2014             self.$search.removeAttr('aria-controls');
2015             self.$search.removeAttr('aria-activedescendant');
2016             self.$search.trigger('focus');
2017         });
2018
2019         container.on('enable', function () {
2020             self.$search.prop('disabled', false);
2021
2022             self._transferTabIndex();
2023         });
2024
2025         container.on('disable', function () {

```

```

2026     self $search prop! 'disabled', true);
2027 });
2028
2029 container on! 'focus', function (evt) {
2030     self $search trigger! 'focus';
2031 });
2032
2033 container on! 'results:focus', function (params) {
2034     if params data _resultId {
2035         self $search attr! 'aria-activedescendant', params data _resultId;
2036     } else {
2037         self $search removeAttr! 'aria-activedescendant';
2038     }
2039 });
2040
2041 this $selection on! 'focusin', '.select2-search--inline', function (evt) {
2042     self trigger! 'focus', evt;
2043 });
2044
2045 this $selection on! 'focusout', '.select2-search--inline', function (evt) {
2046     self _handleBlur! evt;
2047 });
2048
2049 this $selection on! 'keydown', '.select2-search--inline', function (evt) {
2050     evt stopPropagation();
2051
2052     self trigger! 'keypress', evt;
2053
2054     self _keyUpPrevented = evt isDefaultPrevented();
2055
2056     var key = evt which;
2057
2058     if (key === KEYS BACKSPACE && self $search val() === "") {
2059         var $previousChoice = self $searchContainer
2060             .prev! '.select2-selection__choice';
2061
2062         if ($previousChoice.length > 0) {
2063             var item = Utils GetData! $previousChoice[0], 'data';
2064
2065             self searchRemoveChoice! item;
2066
2067             evt preventDefault();
2068         }
2069     }
2070 });
2071
2072 this $selection on! 'click', '.select2-search--inline', function (evt) {
2073     if (self $search val()) {
2074         evt stopPropagation();
2075     }
2076 });
2077
2078 // Try to detect the IE version should the `documentMode` property that
2079 // is stored on the document. This is only implemented in IE and is
2080 // slightly cleaner than doing a user agent check.
2081 // This property is not available in Edge, but Edge also doesn't have
2082 // this bug.
2083 var msie = document documentMode
2084 var disableInputEvents = msie && msie <= 11;
2085
2086 // Workaround for browsers which do not support the `input` event
2087 // This will prevent double-triggering of events for browsers which support
2088 // both the `keyup` and `input` events.
2089 this $selection on!
2090     'input.searchcheck',
2091     '.select2-search--inline',
2092     function (evt) {
2093         // IE will trigger the `input` event when a placeholder is used on a
2094         // search box. To get around this issue, we are forced to ignore all
2095         // `input` events in IE and keep using `keyup`.
2096         if (disableInputEvents) {
2097             self $selection off! ('input.search input.searchcheck');
2098             return;
2099         }
2100
2101         // Unbind the duplicated `keyup` event
2102         self $selection off! ('keyup.search');
2103     }
2104 );
2105
2106 this $selection on!
2107     'keyup.search input.search',
2108     '.select2-search--inline',
2109     function (evt) {
2110         // IE will trigger the `input` event when a placeholder is used on a
2111         // search box. To get around this issue, we are forced to ignore all
2112         // `input` events in IE and keep using `keyup`.
2113         if (disableInputEvents && evt type === 'input') {
2114             self $selection off! ('input.search input.searchcheck');
2115             return;
2116         }
2117
2118         var key = evt which;
2119
2120         // We can freely ignore events from modifier keys
2121         if (key === KEYS SHIFT || key === KEYS CTRL || key === KEYS ALT) {
2122             return;
2123         }
2124
2125         // Tabbing will be handled during the `keydown` phase
2126         if (key === KEYS TAB) {
2127             return;
2128         }
2129
2130         self handleSearch! evt;
2131     }
2132 );
2133 };
2134
2135 /**
2136  * This method will transfer the tabindex attribute from the rendered
2137  * selection to the search box. This allows for the search box to be used as
2138  * the primary focus instead of the selection container.
2139  *
2140  * @private

```

```

2141 */
2142 Search.prototype.transferTabIndex = function (decorated) {
2143   this.$search.attr('tabindex', this.$selection.attr('tabindex'));
2144   this.$selection.attr('tabindex', '-1');
2145 };
2146
2147 Search.prototype.createPlaceholder = function (decorated, placeholder) {
2148   this.$search.attr('placeholder', placeholder.text);
2149 };
2150
2151 Search.prototype.update = function (decorated, data) {
2152   var searchHadFocus = this.$search[0] === document.activeElement;
2153
2154   this.$search.attr('placeholder', "");
2155
2156   decorated.call(this, data);
2157
2158   this.$selection.find('.select2-selection__rendered')
2159     .append(this.$searchContainer);
2160
2161   this.resizeSearch();
2162   if (searchHadFocus) {
2163     this.$search.trigger('focus');
2164   }
2165 };
2166
2167 Search.prototype.handleSearch = function () {
2168   this.resizeSearch();
2169
2170   if (!this._keyUpPrevented) {
2171     var input = this.$search.val();
2172
2173     this.trigger('query', {
2174       term: input
2175     });
2176   }
2177
2178   this._keyUpPrevented = false;
2179 };
2180
2181 Search.prototype.searchRemoveChoice = function (decorated, item) {
2182   this.trigger('unselect', {
2183     data: item
2184   });
2185
2186   this.$search.val(item.text);
2187   this.handleSearch();
2188 };
2189
2190 Search.prototype.resizeSearch = function () {
2191   this.$search.css('width', '25px');
2192
2193   var width = "";
2194
2195   if (this.$search.attr('placeholder') !== "") {
2196     width = this.$selection.find('.select2-selection__rendered').width();
2197   } else {
2198     var minimumWidth = this.$search.val().length + 1;
2199
2200     width = (minimumWidth * 0.75) + 'em';
2201   }
2202
2203   this.$search.css('width', width);
2204 };
2205
2206 return Search;
2207 });
2208
2209 S2.define('select2/selection/eventRelay',[
2210   'jquery'
2211 ], function ($) {
2212   function EventRelay () { }
2213
2214   EventRelay.prototype.bind = function (decorated, container, $container) {
2215     var self = this;
2216     var relayEvents = [
2217       'open', 'opening',
2218       'close', 'closing',
2219       'select', 'selecting',
2220       'unselect', 'unselecting',
2221       'clear', 'clearing'
2222     ];
2223
2224     var preventableEvents = [
2225       'opening', 'closing', 'selecting', 'unselecting', 'clearing'
2226     ];
2227
2228     decorated.call(this, container, $container);
2229
2230     container.on('**', function (name, params) {
2231       // Ignore events that should not be relayed
2232       if ($.inArray(name, relayEvents) === -1) {
2233         return;
2234       }
2235
2236       // The parameters should always be an object
2237       params = params || {};
2238
2239       // Generate the jQuery event for the Select2 event
2240       var evt = $.Event('select2:' + name, {
2241         params: params
2242       });
2243
2244       self.$element.trigger(evt);
2245
2246       // Only handle preventable events if it was one
2247       if ($.inArray(name, preventableEvents) === -1) {
2248         return;
2249       }
2250
2251       params.prevented = evt.isDefaultPrevented();
2252     });
2253   };
2254
2255   return EventRelay;

```

```

2256 });
2257
2258 S2 define('select2/translation',[
2259   'jquery',
2260   'require'
2261 ], function ($, require) {
2262   function Translation (dict) {
2263     this dict = dict || {};
2264   }
2265
2266   Translation.prototype all = function () {
2267     return this dict;
2268   };
2269
2270   Translation.prototype get = function (key) {
2271     return this dict[key];
2272   };
2273
2274   Translation.prototype extend = function (translation) {
2275     this dict = $.extend({}, translation.all(), this dict);
2276   };
2277
2278   // Static functions
2279
2280   Translation._cache = {};
2281
2282   Translation.loadPath = function (path) {
2283     if (!(path in Translation._cache)) {
2284       var translations = require(path);
2285
2286       Translation._cache[path] = translations;
2287     }
2288
2289     return new Translation(Translation._cache[path]);
2290   };
2291
2292   return Translation;
2293 });
2294
2295 S2 define('select2/diacritics',[
2296   function () {
2297     var diacritics = {
2298       '\u24B6': 'A',
2299       '\uFF21': 'A',
2300       '\u00C0': 'A',
2301       '\u00C1': 'A',
2302       '\u00C2': 'A',
2303       '\u1EA6': 'A',
2304       '\u1EA4': 'A',
2305       '\u1EAA': 'A',
2306       '\u1EA8': 'A',
2307       '\u00C3': 'A',
2308       '\u0100': 'A',
2309       '\u0102': 'A',
2310       '\u1EB0': 'A',
2311       '\u1EAE': 'A',
2312       '\u1EB4': 'A',
2313       '\u1EB2': 'A',
2314       '\u0226': 'A',
2315       '\u01E0': 'A',
2316       '\u00C4': 'A',
2317       '\u01DE': 'A',
2318       '\u1EA2': 'A',
2319       '\u00C5': 'A',
2320       '\u01FA': 'A',
2321       '\u01CD': 'A',
2322       '\u0200': 'A',
2323       '\u0202': 'A',
2324       '\u1EA0': 'A',
2325       '\u1EAC': 'A',
2326       '\u1EB6': 'A',
2327       '\u1E00': 'A',
2328       '\u0104': 'A',
2329       '\u023A': 'A',
2330       '\u2C6F': 'A',
2331       '\uA732': 'AA',
2332       '\u00C6': 'AE',
2333       '\u01FC': 'AE',
2334       '\u01E2': 'AE',
2335       '\uA734': 'AO',
2336       '\uA736': 'AU',
2337       '\uA738': 'AV',
2338       '\uA73A': 'AV',
2339       '\uA73C': 'AY',
2340       '\u24B7': 'B',
2341       '\uFF22': 'B',
2342       '\u1E02': 'B',
2343       '\u1E04': 'B',
2344       '\u1E06': 'B',
2345       '\u0243': 'B',
2346       '\u0182': 'B',
2347       '\u0181': 'B',
2348       '\u24B8': 'C',
2349       '\uFF23': 'C',
2350       '\u0106': 'C',
2351       '\u0108': 'C',
2352       '\u010A': 'C',
2353       '\u010C': 'C',
2354       '\u00C7': 'C',
2355       '\u1E08': 'C',
2356       '\u0187': 'C',
2357       '\u023B': 'C',
2358       '\uA73E': 'C',
2359       '\u24B9': 'D',
2360       '\uFF24': 'D',
2361       '\u1E0A': 'D',
2362       '\u010E': 'D',
2363       '\u1E0C': 'D',
2364       '\u1E10': 'D',
2365       '\u1E12': 'D',
2366       '\u1E0E': 'D',
2367       '\u0110': 'D',
2368       '\u018B': 'D',
2369       '\u018A': 'D',
2370

```

2371 '\u0189': 'D',  
2372 '\u0179': 'D',  
2373 '\u01F1': 'DZ',  
2374 '\u01C4': 'DZ',  
2375 '\u01F2': 'Dz',  
2376 '\u01C5': 'Dz',  
2377 '\u0248A': 'E',  
2378 '\u01F25': 'E',  
2379 '\u00C8': 'E',  
2380 '\u00C9': 'E',  
2381 '\u00CA': 'E',  
2382 '\u01EC0': 'E',  
2383 '\u01EBE': 'E',  
2384 '\u01EC4': 'E',  
2385 '\u01EC2': 'E',  
2386 '\u01EBC': 'E',  
2387 '\u0112': 'E',  
2388 '\u01E14': 'E',  
2389 '\u01E16': 'E',  
2390 '\u0114': 'E',  
2391 '\u0116': 'E',  
2392 '\u00C8': 'E',  
2393 '\u01EBA': 'E',  
2394 '\u011A': 'E',  
2395 '\u0204': 'E',  
2396 '\u0206': 'E',  
2397 '\u01EB8': 'E',  
2398 '\u01EC6': 'E',  
2399 '\u0228': 'E',  
2400 '\u01E1C': 'E',  
2401 '\u0118': 'E',  
2402 '\u01E18': 'E',  
2403 '\u01E1A': 'E',  
2404 '\u0190': 'E',  
2405 '\u018E': 'E',  
2406 '\u0248B': 'F',  
2407 '\u01F26': 'F',  
2408 '\u01E1E': 'F',  
2409 '\u0191': 'F',  
2410 '\u0177B': 'F',  
2411 '\u0248C': 'G',  
2412 '\u01F27': 'G',  
2413 '\u01F4': 'G',  
2414 '\u011C': 'G',  
2415 '\u01E20': 'G',  
2416 '\u011E': 'G',  
2417 '\u0120': 'G',  
2418 '\u01E6': 'G',  
2419 '\u0122': 'G',  
2420 '\u01E4': 'G',  
2421 '\u0193': 'G',  
2422 '\u017A0': 'G',  
2423 '\u0177D': 'G',  
2424 '\u0177E': 'G',  
2425 '\u0248D': 'H',  
2426 '\u01F28': 'H',  
2427 '\u0124': 'H',  
2428 '\u01E22': 'H',  
2429 '\u01E26': 'H',  
2430 '\u021E': 'H',  
2431 '\u01E24': 'H',  
2432 '\u01E28': 'H',  
2433 '\u01E2A': 'H',  
2434 '\u0126': 'H',  
2435 '\u02C67': 'H',  
2436 '\u02C75': 'H',  
2437 '\u0178D': 'H',  
2438 '\u0248E': 'I',  
2439 '\u01F29': 'I',  
2440 '\u00CC': 'I',  
2441 '\u00CD': 'I',  
2442 '\u00CE': 'I',  
2443 '\u0128': 'I',  
2444 '\u012A': 'I',  
2445 '\u012C': 'I',  
2446 '\u0130': 'I',  
2447 '\u00CF': 'I',  
2448 '\u01E2E': 'I',  
2449 '\u01EC8': 'I',  
2450 '\u01CF': 'I',  
2451 '\u0208': 'I',  
2452 '\u020A': 'I',  
2453 '\u01ECA': 'I',  
2454 '\u012E': 'I',  
2455 '\u01E2C': 'I',  
2456 '\u0197': 'I',  
2457 '\u0248F': 'J',  
2458 '\u01F2A': 'J',  
2459 '\u0134': 'J',  
2460 '\u0248': 'J',  
2461 '\u024C0': 'K',  
2462 '\u01F2B': 'K',  
2463 '\u01E30': 'K',  
2464 '\u01E8': 'K',  
2465 '\u01E32': 'K',  
2466 '\u0136': 'K',  
2467 '\u01E34': 'K',  
2468 '\u0198': 'K',  
2469 '\u02C69': 'K',  
2470 '\u017A0': 'K',  
2471 '\u017A2': 'K',  
2472 '\u017A4': 'K',  
2473 '\u017A2': 'K',  
2474 '\u024C1': 'L',  
2475 '\u01F2C': 'L',  
2476 '\u013F': 'L',  
2477 '\u0139': 'L',  
2478 '\u013D': 'L',  
2479 '\u01E36': 'L',  
2480 '\u01E38': 'L',  
2481 '\u013B': 'L',  
2482 '\u01E3C': 'L',  
2483 '\u01E3A': 'L',  
2484 '\u0141': 'L',  
2485 '\u023D': 'L',



2486 '\u2C62': 'L',  
2487 '\u2C60': 'L',  
2488 '\uA748': 'L',  
2489 '\uA746': 'L',  
2490 '\uA780': 'L',  
2491 '\u01C7': 'L',  
2492 '\u01C8': 'L',  
2493 '\u24C2': 'M',  
2494 '\uFF2D': 'M',  
2495 '\u1E3E': 'M',  
2496 '\u1E40': 'M',  
2497 '\u1E42': 'M',  
2498 '\u2C6E': 'M',  
2499 '\u019C': 'M',  
2500 '\u24C3': 'N',  
2501 '\uFF2E': 'N',  
2502 '\u01F8': 'N',  
2503 '\u0143': 'N',  
2504 '\u00D1': 'N',  
2505 '\u1E44': 'N',  
2506 '\u0147': 'N',  
2507 '\u1E46': 'N',  
2508 '\u0145': 'N',  
2509 '\u1E4A': 'N',  
2510 '\u1E48': 'N',  
2511 '\u0220': 'N',  
2512 '\u019D': 'N',  
2513 '\uA790': 'N',  
2514 '\uA7A4': 'N',  
2515 '\u01CA': 'N',  
2516 '\u01CB': 'N',  
2517 '\u24C4': 'O',  
2518 '\uFF2F': 'O',  
2519 '\u00D2': 'O',  
2520 '\u00D3': 'O',  
2521 '\u00D4': 'O',  
2522 '\u1ED2': 'O',  
2523 '\u1ED0': 'O',  
2524 '\u1ED6': 'O',  
2525 '\u1ED4': 'O',  
2526 '\u00D5': 'O',  
2527 '\u1E4C': 'O',  
2528 '\u022C': 'O',  
2529 '\u1E4E': 'O',  
2530 '\u014C': 'O',  
2531 '\u1E50': 'O',  
2532 '\u1E52': 'O',  
2533 '\u014E': 'O',  
2534 '\u022E': 'O',  
2535 '\u0230': 'O',  
2536 '\u00D6': 'O',  
2537 '\u022A': 'O',  
2538 '\u1ECE': 'O',  
2539 '\u0150': 'O',  
2540 '\u01D1': 'O',  
2541 '\u020C': 'O',  
2542 '\u020E': 'O',  
2543 '\u01A0': 'O',  
2544 '\u1EDC': 'O',  
2545 '\u1EDA': 'O',  
2546 '\u1EE0': 'O',  
2547 '\u1EDE': 'O',  
2548 '\u1EE2': 'O',  
2549 '\u1ECC': 'O',  
2550 '\u1ED8': 'O',  
2551 '\u01EA': 'O',  
2552 '\u01EC': 'O',  
2553 '\u00D8': 'O',  
2554 '\u01FE': 'O',  
2555 '\u0186': 'O',  
2556 '\u019F': 'O',  
2557 '\uA74A': 'O',  
2558 '\uA74C': 'O',  
2559 '\u0152': 'OE',  
2560 '\u01A2': 'OI',  
2561 '\uA74E': 'OO',  
2562 '\u0222': 'OU',  
2563 '\u24C5': 'P',  
2564 '\uFF30': 'P',  
2565 '\u1E54': 'P',  
2566 '\u1E56': 'P',  
2567 '\u01A4': 'P',  
2568 '\u2C63': 'P',  
2569 '\uA750': 'P',  
2570 '\uA752': 'P',  
2571 '\uA754': 'P',  
2572 '\u24C6': 'Q',  
2573 '\uFF31': 'Q',  
2574 '\uA756': 'Q',  
2575 '\uA758': 'Q',  
2576 '\u024A': 'Q',  
2577 '\u24C7': 'R',  
2578 '\uFF32': 'R',  
2579 '\u0154': 'R',  
2580 '\u1E58': 'R',  
2581 '\u0158': 'R',  
2582 '\u0210': 'R',  
2583 '\u0212': 'R',  
2584 '\u1E5A': 'R',  
2585 '\u1E5C': 'R',  
2586 '\u0156': 'R',  
2587 '\u1E5E': 'R',  
2588 '\u024C': 'R',  
2589 '\u2C64': 'R',  
2590 '\uA75A': 'R',  
2591 '\uA7A6': 'R',  
2592 '\uA782': 'R',  
2593 '\u24C8': 'S',  
2594 '\uFF33': 'S',  
2595 '\u1E9E': 'S',  
2596 '\u015A': 'S',  
2597 '\u1E64': 'S',  
2598 '\u015C': 'S',  
2599 '\u1E60': 'S',  
2600 '\u0160': 'S',

2601 '\u1E66': 'S',  
2602 '\u1E62': 'S',  
2603 '\u1E68': 'S',  
2604 '\u0218': 'S',  
2605 '\u015E': 'S',  
2606 '\u2C7E': 'S',  
2607 '\uA7A8': 'S',  
2608 '\uA784': 'S',  
2609 '\u24C9': 'T',  
2610 '\uFF34': 'T',  
2611 '\u1E6A': 'T',  
2612 '\u0164': 'T',  
2613 '\u1E6C': 'T',  
2614 '\u021A': 'T',  
2615 '\u0162': 'T',  
2616 '\u1E70': 'T',  
2617 '\u1E6E': 'T',  
2618 '\u0166': 'T',  
2619 '\u01AC': 'T',  
2620 '\u01AE': 'T',  
2621 '\u023E': 'T',  
2622 '\uA786': 'T',  
2623 '\uA728': 'TZ',  
2624 '\u24CA': 'U',  
2625 '\uFF35': 'U',  
2626 '\u00D9': 'U',  
2627 '\u00DA': 'U',  
2628 '\u00DB': 'U',  
2629 '\u0168': 'U',  
2630 '\u1E78': 'U',  
2631 '\u016A': 'U',  
2632 '\u1E7A': 'U',  
2633 '\u016C': 'U',  
2634 '\u00DC': 'U',  
2635 '\u01DB': 'U',  
2636 '\u01D7': 'U',  
2637 '\u01D5': 'U',  
2638 '\u01D9': 'U',  
2639 '\u1EE6': 'U',  
2640 '\u016E': 'U',  
2641 '\u0170': 'U',  
2642 '\u01D3': 'U',  
2643 '\u0214': 'U',  
2644 '\u0216': 'U',  
2645 '\u01AF': 'U',  
2646 '\u1EEA': 'U',  
2647 '\u1EE8': 'U',  
2648 '\u1EEE': 'U',  
2649 '\u1EEC': 'U',  
2650 '\u1EF0': 'U',  
2651 '\u1EE4': 'U',  
2652 '\u1E72': 'U',  
2653 '\u0172': 'U',  
2654 '\u1E76': 'U',  
2655 '\u1E74': 'U',  
2656 '\u0244': 'U',  
2657 '\u24CB': 'V',  
2658 '\uFF36': 'V',  
2659 '\u1E7C': 'V',  
2660 '\u1E7E': 'V',  
2661 '\u01B2': 'V',  
2662 '\uA75E': 'V',  
2663 '\u0245': 'V',  
2664 '\uA760': 'VY',  
2665 '\u24CC': 'W',  
2666 '\uFF37': 'W',  
2667 '\u1E80': 'W',  
2668 '\u1E82': 'W',  
2669 '\u0174': 'W',  
2670 '\u1E86': 'W',  
2671 '\u1E84': 'W',  
2672 '\u1E88': 'W',  
2673 '\u2C72': 'W',  
2674 '\u24CD': 'X',  
2675 '\uFF38': 'X',  
2676 '\u1E8A': 'X',  
2677 '\u1E8C': 'X',  
2678 '\u24CE': 'Y',  
2679 '\uFF39': 'Y',  
2680 '\u1EF2': 'Y',  
2681 '\u00DD': 'Y',  
2682 '\u0176': 'Y',  
2683 '\u1EF8': 'Y',  
2684 '\u0232': 'Y',  
2685 '\u1E8E': 'Y',  
2686 '\u0178': 'Y',  
2687 '\u1EF6': 'Y',  
2688 '\u1EF4': 'Y',  
2689 '\u01B3': 'Y',  
2690 '\u024E': 'Y',  
2691 '\u1EFE': 'Y',  
2692 '\u24CF': 'Z',  
2693 '\uFF3A': 'Z',  
2694 '\u0179': 'Z',  
2695 '\u1E90': 'Z',  
2696 '\u017B': 'Z',  
2697 '\u017D': 'Z',  
2698 '\u1E92': 'Z',  
2699 '\u1E94': 'Z',  
2700 '\u01B5': 'Z',  
2701 '\u0224': 'Z',  
2702 '\u2C7F': 'Z',  
2703 '\u2C68': 'Z',  
2704 '\uA762': 'Z',  
2705 '\u24D0': 'a',  
2706 '\uFF41': 'a',  
2707 '\u1E9A': 'a',  
2708 '\u00E0': 'a',  
2709 '\u00E1': 'a',  
2710 '\u00E2': 'a',  
2711 '\u1EA7': 'a',  
2712 '\u1EA5': 'a',  
2713 '\u1EAB': 'a',  
2714 '\u1EA9': 'a',  
2715 '\u00E3': 'a',

2716 'u0101': 'a',  
2717 'u0103': 'a',  
2718 'u1EB1': 'a',  
2719 'u1EAF': 'a',  
2720 'u1EB5': 'a',  
2721 'u1EB3': 'a',  
2722 'u0227': 'a',  
2723 'u01E1': 'a',  
2724 'u00E4': 'a',  
2725 'u01DF': 'a',  
2726 'u1EA3': 'a',  
2727 'u00E5': 'a',  
2728 'u01FB': 'a',  
2729 'u01CE': 'a',  
2730 'u0201': 'a',  
2731 'u0203': 'a',  
2732 'u1EA1': 'a',  
2733 'u1EAD': 'a',  
2734 'u1EB7': 'a',  
2735 'u1E01': 'a',  
2736 'u0105': 'a',  
2737 'u2C65': 'a',  
2738 'u0250': 'a',  
2739 'uA733': 'aa',  
2740 'u00E6': 'ae',  
2741 'u01FD': 'ae',  
2742 'u01E3': 'ae',  
2743 'uA735': 'ao',  
2744 'uA737': 'au',  
2745 'uA739': 'av',  
2746 'uA73B': 'av',  
2747 'uA73D': 'ay',  
2748 'u24D1': 'b',  
2749 'uFF42': 'b',  
2750 'u1E03': 'b',  
2751 'u1E05': 'b',  
2752 'u1E07': 'b',  
2753 'u0180': 'b',  
2754 'u0183': 'b',  
2755 'u0253': 'b',  
2756 'u24D2': 'c',  
2757 'uFF43': 'c',  
2758 'u0107': 'c',  
2759 'u0109': 'c',  
2760 'u010B': 'c',  
2761 'u010D': 'c',  
2762 'u00E7': 'c',  
2763 'u1E09': 'c',  
2764 'u0188': 'c',  
2765 'u023C': 'c',  
2766 'uA73F': 'c',  
2767 'u2184': 'c',  
2768 'u24D3': 'd',  
2769 'uFF44': 'd',  
2770 'u1E0B': 'd',  
2771 'u010F': 'd',  
2772 'u1E0D': 'd',  
2773 'u1E11': 'd',  
2774 'u1E13': 'd',  
2775 'u1E0F': 'd',  
2776 'u0111': 'd',  
2777 'u018C': 'd',  
2778 'u0256': 'd',  
2779 'u0257': 'd',  
2780 'uA77A': 'd',  
2781 'u01F3': 'dz',  
2782 'u01C6': 'dz',  
2783 'u24D4': 'e',  
2784 'uFF45': 'e',  
2785 'u00E8': 'e',  
2786 'u00E9': 'e',  
2787 'u00EA': 'e',  
2788 'u1EC1': 'e',  
2789 'u1EBF': 'e',  
2790 'u1EC5': 'e',  
2791 'u1EC3': 'e',  
2792 'u1EBD': 'e',  
2793 'u0113': 'e',  
2794 'u1E15': 'e',  
2795 'u1E17': 'e',  
2796 'u0115': 'e',  
2797 'u0117': 'e',  
2798 'u00EB': 'e',  
2799 'u1EBB': 'e',  
2800 'u011B': 'e',  
2801 'u0205': 'e',  
2802 'u0207': 'e',  
2803 'u1EB9': 'e',  
2804 'u1EC7': 'e',  
2805 'u0229': 'e',  
2806 'u1E1D': 'e',  
2807 'u0119': 'e',  
2808 'u1E19': 'e',  
2809 'u1E1B': 'e',  
2810 'u0247': 'e',  
2811 'u025B': 'e',  
2812 'u01DD': 'e',  
2813 'u24D5': 'f',  
2814 'uFF46': 'f',  
2815 'u1E1F': 'f',  
2816 'u0192': 'f',  
2817 'uA77C': 'f',  
2818 'u24D6': 'g',  
2819 'uFF47': 'g',  
2820 'u01F5': 'g',  
2821 'u011D': 'g',  
2822 'u1E21': 'g',  
2823 'u011F': 'g',  
2824 'u0121': 'g',  
2825 'u01E7': 'g',  
2826 'u0123': 'g',  
2827 'u01E5': 'g',  
2828 'u0260': 'g',  
2829 'uA7A1': 'g',  
2830 'u1D79': 'g',

2831 'uA77F': 'g',  
2832 'u24D7': 'h',  
2833 'uFF48': 'h',  
2834 'u0125': 'h',  
2835 'u1E23': 'h',  
2836 'u1E27': 'h',  
2837 'u021F': 'h',  
2838 'u1E25': 'h',  
2839 'u1E29': 'h',  
2840 'u1E28': 'h',  
2841 'u1E96': 'h',  
2842 'u0127': 'h',  
2843 'u2C68': 'h',  
2844 'u2C76': 'h',  
2845 'u0265': 'h',  
2846 'u0195': 'hw',  
2847 'u24D8': 'i',  
2848 'uFF49': 'i',  
2849 'u00EC': 'i',  
2850 'u00ED': 'i',  
2851 'u00EE': 'i',  
2852 'u0129': 'i',  
2853 'u012B': 'i',  
2854 'u012D': 'i',  
2855 'u00EF': 'i',  
2856 'u1E2F': 'i',  
2857 'u1EC9': 'i',  
2858 'u01D0': 'i',  
2859 'u0209': 'i',  
2860 'u020B': 'i',  
2861 'u1ECB': 'i',  
2862 'u012F': 'i',  
2863 'u1E2D': 'i',  
2864 'u0268': 'i',  
2865 'u0131': 'i',  
2866 'u24D9': 'j',  
2867 'uFF4A': 'j',  
2868 'u0135': 'j',  
2869 'u01F0': 'j',  
2870 'u0249': 'j',  
2871 'u24DA': 'k',  
2872 'uFF4B': 'k',  
2873 'u1E31': 'k',  
2874 'u01E9': 'k',  
2875 'u1E33': 'k',  
2876 'u0137': 'k',  
2877 'u1E35': 'k',  
2878 'u0199': 'k',  
2879 'u2C6A': 'k',  
2880 'uA741': 'k',  
2881 'uA743': 'k',  
2882 'uA745': 'k',  
2883 'uA7A3': 'k',  
2884 'u24DB': 'l',  
2885 'uFF4C': 'l',  
2886 'u0140': 'l',  
2887 'u013A': 'l',  
2888 'u013E': 'l',  
2889 'u1E37': 'l',  
2890 'u1E39': 'l',  
2891 'u013C': 'l',  
2892 'u1E3D': 'l',  
2893 'u1E3B': 'l',  
2894 'u017F': 'l',  
2895 'u0142': 'l',  
2896 'u019A': 'l',  
2897 'u026B': 'l',  
2898 'u2C61': 'l',  
2899 'uA749': 'l',  
2900 'uA781': 'l',  
2901 'uA747': 'l',  
2902 'u01C9': 'lj',  
2903 'u24DC': 'm',  
2904 'uFF4D': 'm',  
2905 'u1E3F': 'm',  
2906 'u1E41': 'm',  
2907 'u1E43': 'm',  
2908 'u0271': 'm',  
2909 'u026F': 'm',  
2910 'u24DD': 'n',  
2911 'uFF4E': 'n',  
2912 'u01F9': 'n',  
2913 'u0144': 'n',  
2914 'u00F1': 'n',  
2915 'u1E45': 'n',  
2916 'u0148': 'n',  
2917 'u1E47': 'n',  
2918 'u0146': 'n',  
2919 'u1E4B': 'n',  
2920 'u1E49': 'n',  
2921 'u019E': 'n',  
2922 'u0272': 'n',  
2923 'u0149': 'n',  
2924 'uA791': 'n',  
2925 'uA7A5': 'n',  
2926 'u01CC': 'nj',  
2927 'u24DE': 'o',  
2928 'uFF4F': 'o',  
2929 'u00F2': 'o',  
2930 'u00F3': 'o',  
2931 'u00F4': 'o',  
2932 'u1ED3': 'o',  
2933 'u1ED1': 'o',  
2934 'u1ED7': 'o',  
2935 'u1ED5': 'o',  
2936 'u00F5': 'o',  
2937 'u1E4D': 'o',  
2938 'u022D': 'o',  
2939 'u1E4F': 'o',  
2940 'u014D': 'o',  
2941 'u1E51': 'o',  
2942 'u1E53': 'o',  
2943 'u014F': 'o',  
2944 'u022F': 'o',  
2945 'u0231': 'o',

2946 'ʷ00F6': 'o',  
2947 'ʷ022B': 'o',  
2948 'ʷ1ECF': 'o',  
2949 'ʷ0151': 'o',  
2950 'ʷ01D2': 'o',  
2951 'ʷ020D': 'o',  
2952 'ʷ020F': 'o',  
2953 'ʷ01A1': 'o',  
2954 'ʷ1EDD': 'o',  
2955 'ʷ1EDB': 'o',  
2956 'ʷ1EE1': 'o',  
2957 'ʷ1EDF': 'o',  
2958 'ʷ1EE3': 'o',  
2959 'ʷ1ECD': 'o',  
2960 'ʷ1ED9': 'o',  
2961 'ʷ01EB': 'o',  
2962 'ʷ01ED': 'o',  
2963 'ʷ00F8': 'o',  
2964 'ʷ01FF': 'o',  
2965 'ʷ0254': 'o',  
2966 'ʷA748': 'o',  
2967 'ʷA74D': 'o',  
2968 'ʷ0275': 'o',  
2969 'ʷ0153': 'oe',  
2970 'ʷ01A3': 'oi',  
2971 'ʷ0223': 'ou',  
2972 'ʷA74F': 'oo',  
2973 'ʷ24DF': 'p',  
2974 'ʷFF50': 'p',  
2975 'ʷ1E55': 'p',  
2976 'ʷ1E57': 'p',  
2977 'ʷ01A5': 'p',  
2978 'ʷ1D7D': 'p',  
2979 'ʷA751': 'p',  
2980 'ʷA753': 'p',  
2981 'ʷA755': 'p',  
2982 'ʷ24E0': 'q',  
2983 'ʷFF51': 'q',  
2984 'ʷ0248': 'q',  
2985 'ʷA757': 'q',  
2986 'ʷA759': 'q',  
2987 'ʷ24E1': 'r',  
2988 'ʷFF52': 'r',  
2989 'ʷ0155': 'r',  
2990 'ʷ1E59': 'r',  
2991 'ʷ0159': 'r',  
2992 'ʷ0211': 'r',  
2993 'ʷ0213': 'r',  
2994 'ʷ1E58': 'r',  
2995 'ʷ1E5D': 'r',  
2996 'ʷ0157': 'r',  
2997 'ʷ1E5F': 'r',  
2998 'ʷ024D': 'r',  
2999 'ʷ027D': 'r',  
3000 'ʷA758': 'r',  
3001 'ʷA7A7': 'r',  
3002 'ʷA783': 'r',  
3003 'ʷ24E2': 's',  
3004 'ʷFF53': 's',  
3005 'ʷ00DF': 's',  
3006 'ʷ015B': 's',  
3007 'ʷ1E65': 's',  
3008 'ʷ015D': 's',  
3009 'ʷ1E61': 's',  
3010 'ʷ0161': 's',  
3011 'ʷ1E67': 's',  
3012 'ʷ1E63': 's',  
3013 'ʷ1E69': 's',  
3014 'ʷ0219': 's',  
3015 'ʷ015F': 's',  
3016 'ʷ023F': 's',  
3017 'ʷA7A9': 's',  
3018 'ʷA785': 's',  
3019 'ʷ1E9B': 's',  
3020 'ʷ24E3': 't',  
3021 'ʷFF54': 't',  
3022 'ʷ1E68': 't',  
3023 'ʷ1E97': 't',  
3024 'ʷ0165': 't',  
3025 'ʷ1E6D': 't',  
3026 'ʷ021B': 't',  
3027 'ʷ0163': 't',  
3028 'ʷ1E71': 't',  
3029 'ʷ1E6F': 't',  
3030 'ʷ0167': 't',  
3031 'ʷ01AD': 't',  
3032 'ʷ0288': 't',  
3033 'ʷ2C66': 't',  
3034 'ʷA787': 't',  
3035 'ʷA729': 'tz',  
3036 'ʷ24E4': 'u',  
3037 'ʷFF55': 'u',  
3038 'ʷ00F9': 'u',  
3039 'ʷ00FA': 'u',  
3040 'ʷ00FB': 'u',  
3041 'ʷ0169': 'u',  
3042 'ʷ1E79': 'u',  
3043 'ʷ016B': 'u',  
3044 'ʷ1E7B': 'u',  
3045 'ʷ016D': 'u',  
3046 'ʷ00FC': 'u',  
3047 'ʷ01DC': 'u',  
3048 'ʷ01D8': 'u',  
3049 'ʷ01D6': 'u',  
3050 'ʷ01DA': 'u',  
3051 'ʷ1EE7': 'u',  
3052 'ʷ016F': 'u',  
3053 'ʷ0171': 'u',  
3054 'ʷ01D4': 'u',  
3055 'ʷ0215': 'u',  
3056 'ʷ0217': 'u',  
3057 'ʷ01B0': 'u',  
3058 'ʷ1EEB': 'u',  
3059 'ʷ1EE9': 'u',  
3060 'ʷ1EEF': 'u',

```

3061 '\u1EED': 'u',
3062 '\u1EF1': 'u',
3063 '\u1EE5': 'u',
3064 '\u1E73': 'u',
3065 '\u0173': 'u',
3066 '\u1E77': 'u',
3067 '\u1E75': 'u',
3068 '\u0289': 'u',
3069 '\u24E5': 'v',
3070 '\uFF56': 'v',
3071 '\u1E7D': 'v',
3072 '\u1E7F': 'v',
3073 '\u028B': 'v',
3074 '\uA75F': 'v',
3075 '\u028C': 'v',
3076 '\uA761': 'vy',
3077 '\u24E6': 'w',
3078 '\uFF57': 'w',
3079 '\u1E81': 'w',
3080 '\u1E83': 'w',
3081 '\u0175': 'w',
3082 '\u1E87': 'w',
3083 '\u1E85': 'w',
3084 '\u1E98': 'w',
3085 '\u1E89': 'w',
3086 '\u2C73': 'w',
3087 '\u24E7': 'x',
3088 '\uFF58': 'x',
3089 '\u1E8B': 'x',
3090 '\u1E8D': 'x',
3091 '\u24E8': 'y',
3092 '\uFF59': 'y',
3093 '\u1EF3': 'y',
3094 '\u00FD': 'y',
3095 '\u0177': 'y',
3096 '\u1EF9': 'y',
3097 '\u0233': 'y',
3098 '\u1E8F': 'y',
3099 '\u00FF': 'y',
3100 '\u1EF7': 'y',
3101 '\u1E99': 'y',
3102 '\u1EF5': 'y',
3103 '\u01B4': 'y',
3104 '\u024F': 'y',
3105 '\u1EFF': 'y',
3106 '\u24E9': 'z',
3107 '\uFF5A': 'z',
3108 '\u017A': 'z',
3109 '\u1E91': 'z',
3110 '\u017C': 'z',
3111 '\u017E': 'z',
3112 '\u1E93': 'z',
3113 '\u1E95': 'z',
3114 '\u01B6': 'z',
3115 '\u0225': 'z',
3116 '\u0240': 'z',
3117 '\u2C6C': 'z',
3118 '\uA763': 'z',
3119 '\u0386': '\u0391',
3120 '\u0388': '\u0395',
3121 '\u0389': '\u0397',
3122 '\u038A': '\u0399',
3123 '\u03AA': '\u0399',
3124 '\u038C': '\u039F',
3125 '\u038E': '\u03A5',
3126 '\u03AB': '\u03A5',
3127 '\u038F': '\u03A9',
3128 '\u03AC': '\u03B1',
3129 '\u03AD': '\u03B5',
3130 '\u03AE': '\u03B7',
3131 '\u03AF': '\u03B9',
3132 '\u03CA': '\u03B9',
3133 '\u0390': '\u03B9',
3134 '\u03CC': '\u03BF',
3135 '\u03CD': '\u03C5',
3136 '\u03CB': '\u03C5',
3137 '\u03B0': '\u03C5',
3138 '\u03CE': '\u03C9',
3139 '\u03C2': '\u03C3',
3140 '\u2019': ''
3141 };
3142
3143 return diacritics;
3144 });
3145
3146 S2.define('select2/data/base',[
3147   './utils'
3148 ],function (Utils) {
3149   function BaseAdapter ($element, options) {
3150     BaseAdapter.__super__.constructor.call(this);
3151   }
3152
3153   Utils.Extend(BaseAdapter, Utils.Observable);
3154
3155   BaseAdapter.prototype.current = function (callback) {
3156     throw new Error('The `current` method must be defined in child classes.');
```

```

3176 if (data.id != null) {
3177   id += '-' + data.id.toString();
3178 } else {
3179   id += '-' + Utils.generateChars(4);
3180 }
3181 return id;
3182 };
3183
3184 return BaseAdapter;
3185 });
3186
3187 S2.define('select2/data/select',[
3188   './base',
3189   './utils',
3190   'jquery'
3191 ],function (BaseAdapter, Utils, $) {
3192   function SelectAdapter (element, options) {
3193     this.$element = $element;
3194     this.options = options;
3195
3196     SelectAdapter.__super__.constructor.call(this);
3197   }
3198
3199   Utils.Extend(SelectAdapter, BaseAdapter);
3200
3201   SelectAdapter.prototype.current = function (callback) {
3202     var data = [];
3203     var self = this;
3204
3205     this.$element.find(':selected').each(function () {
3206       var $option = $(this);
3207
3208       var option = self.item($option);
3209
3210       data.push(option);
3211     });
3212
3213     callback(data);
3214   };
3215
3216   SelectAdapter.prototype.select = function (data) {
3217     var self = this;
3218
3219     data.selected = true;
3220
3221     // If data.element is a DOM node, use it instead
3222     if ($ (data.element).is('option')) {
3223       data.element.selected = true;
3224
3225       this.$element.trigger('input').trigger('change');
3226
3227       return;
3228     }
3229
3230     if (this.$element.prop('multiple')) {
3231       this.current(function (currentData) {
3232         var val = [];
3233
3234         data = [data];
3235         data.push.apply(data, currentData);
3236
3237         for (var d = 0; d < data.length; d++) {
3238           var id = data[d].id;
3239
3240           if ($.inArray(id, val) === -1) {
3241             val.push(id);
3242           }
3243         }
3244
3245         self.$element.val(val);
3246         self.$element.trigger('input').trigger('change');
3247       });
3248     } else {
3249       var val = data.id;
3250
3251       this.$element.val(val);
3252       this.$element.trigger('input').trigger('change');
3253     }
3254   };
3255
3256   SelectAdapter.prototype.unselect = function (data) {
3257     var self = this;
3258
3259     if (!this.$element.prop('multiple')) {
3260       return;
3261     }
3262
3263     data.selected = false;
3264
3265     if ($ (data.element).is('option')) {
3266       data.element.selected = false;
3267
3268       this.$element.trigger('input').trigger('change');
3269
3270       return;
3271     }
3272
3273     this.current(function (currentData) {
3274       var val = [];
3275
3276       for (var d = 0; d < currentData.length; d++) {
3277         var id = currentData[d].id;
3278
3279         if (id !== data.id && $.inArray(id, val) === -1) {
3280           val.push(id);
3281         }
3282       }
3283
3284       self.$element.val(val);
3285
3286       self.$element.trigger('input').trigger('change');
3287     });
3288   };
3289
3290   SelectAdapter.prototype.bind = function (container, $container) {

```

```

3291 var self = this;
3292
3293 this.container = container;
3294
3295 container.on('select', function (params) {
3296     self.select(params.data);
3297 });
3298
3299 container.on('unselect', function (params) {
3300     self.unselect(params.data);
3301 });
3302 };
3303
3304 SelectAdapter.prototype.destroy = function () {
3305     // Remove anything added to child elements
3306     this.$element.find('*').each(function () {
3307         // Remove any custom data set by Select2
3308         Utils.RemoveData(this);
3309     });
3310 };
3311
3312 SelectAdapter.prototype.query = function (params, callback) {
3313     var data = [];
3314     var self = this;
3315
3316     var $options = this.$element.children();
3317
3318     $options.each(function () {
3319         var $option = $(this);
3320
3321         if (!$option.is('option') && !$option.is('optgroup')) {
3322             return;
3323         }
3324
3325         var option = self.item($option);
3326
3327         var matches = self.matches(params, option);
3328
3329         if (matches !== null) {
3330             data.push(matches);
3331         }
3332     });
3333
3334     callback({
3335         results: data
3336     });
3337 };
3338
3339 SelectAdapter.prototype.addOptions = function ($options) {
3340     Utils.appendMany(this.$element, $options);
3341 };
3342
3343 SelectAdapter.prototype.option = function (data) {
3344     var option;
3345
3346     if (data.children) {
3347         option = document.createElement('optgroup');
3348         option.label = data.text;
3349     } else {
3350         option = document.createElement('option');
3351
3352         if (option.textContent !== undefined) {
3353             option.textContent = data.text;
3354         } else {
3355             option.innerText = data.text;
3356         }
3357     }
3358
3359     if (data.id !== undefined) {
3360         option.value = data.id;
3361     }
3362
3363     if (data.disabled) {
3364         option.disabled = true;
3365     }
3366
3367     if (data.selected) {
3368         option.selected = true;
3369     }
3370
3371     if (data.title) {
3372         option.title = data.title;
3373     }
3374
3375     var $option = $(option);
3376
3377     var normalizedData = this._normalizeItem(data);
3378     normalizedData.element = option;
3379
3380     // Override the option's data with the combined data
3381     Utils.StoreData(option, 'data', normalizedData);
3382
3383     return $option;
3384 };
3385
3386 SelectAdapter.prototype.item = function ($option) {
3387     var data = {};
3388
3389     data = Utils.GetData($option[0], 'data');
3390
3391     if (data !== null) {
3392         return data;
3393     }
3394
3395     if ($option.is('option')) {
3396         data = {
3397             id: $option.val(),
3398             text: $option.text(),
3399             disabled: $option.prop('disabled'),
3400             selected: $option.prop('selected'),
3401             title: $option.prop('title')
3402         };
3403     } else if ($option.is('optgroup')) {
3404         data = {
3405             text: $option.prop('label'),

```



```

3406     children: [],
3407     title: $option prop('title')
3408   };
3409
3410   var $children = $option children('option');
3411   var children = [];
3412
3413   for (var c = 0, c < $children.length; c++) {
3414     var $child = $($children[c]);
3415
3416     var child = this item.$child;
3417
3418     children.push(child);
3419   }
3420
3421   data children = children;
3422 }
3423
3424 data = this._normalizeItem(data);
3425 data.element = $option[0];
3426
3427 Utils.StoreData($option[0], 'data', data);
3428
3429 return data;
3430 };
3431
3432 SelectAdapter.prototype._normalizeItem = function (item) {
3433   if (item !== Object(item)) {
3434     item = {
3435       id: item,
3436       text: item
3437     };
3438   }
3439
3440   item = $ extend({}, {
3441     text: ""
3442   }, item);
3443
3444   var defaults = {
3445     selected: false,
3446     disabled: false
3447   };
3448
3449   if (item.id !== null) {
3450     item.id = item.id.toString();
3451   }
3452
3453   if (item.text !== null) {
3454     item.text = item.text.toString();
3455   }
3456
3457   if (item._resultId == null && item.id && this.container !== null) {
3458     item._resultId = this.generateResultId(this.container, item);
3459   }
3460
3461   return $ extend({}, defaults, item);
3462 };
3463
3464 SelectAdapter.prototype.matches = function (params, data) {
3465   var matcher = this.options.get('matcher');
3466
3467   return matcher(params, data);
3468 };
3469
3470 return SelectAdapter;
3471 });
3472
3473 S2.define('select2/data/array',[
3474   './select',
3475   './utils',
3476   'jquery'
3477 ], function (SelectAdapter, Utils, $) {
3478   function ArrayAdapter ($element, options) {
3479     this._dataToConvert = options.get('data') || [];
3480
3481     ArrayAdapter.__super__.constructor.call(this, $element, options);
3482   }
3483
3484   Utils.Extend(ArrayAdapter, SelectAdapter);
3485
3486   ArrayAdapter.prototype.bind = function (container, $container) {
3487     ArrayAdapter.__super__.bind.call(this, container, $container);
3488
3489     this.addOptions(this.convertToOptions(this._dataToConvert));
3490   };
3491
3492   ArrayAdapter.prototype.select = function (data) {
3493     var $option = this.$element.find('option').filter(function (i, elm) {
3494       return elm.value == data.id.toString();
3495     });
3496
3497     if ($option.length == 0) {
3498       $option = this.option(data);
3499
3500       this.addOptions($option);
3501     }
3502
3503     ArrayAdapter.__super__.select.call(this, data);
3504   };
3505
3506   ArrayAdapter.prototype.convertToOptions = function (data) {
3507     var self = this;
3508
3509     var $existing = this.$element.find('option');
3510     var existingIds = $existing.map(function () {
3511       return self.item($(this)).id;
3512     }).get();
3513
3514     var $options = [];
3515
3516     // Filter out all items except for the one passed in the argument
3517     function onlyItem (item) {
3518       return function () {
3519         return $(this).val() == item.id;
3520       };
3521     }

```

```

3521 }
3522
3523 for (var d = 0; d < data.length; d++) {
3524     var item = this._normalizeItem(data[d]);
3525
3526     // Skip items which were pre-loaded, only merge the data
3527     if (!$.isArray(item.id) || existingIds.indexOf(item.id) >= 0) {
3528         var $existingOption = $existing.filter(function(option) {
3529             return option.data('id') === item.id;
3530         });
3531         var existingData = this._getItemData($existingOption);
3532         var newData = $.extend(true, {}, item, existingData);
3533         var $newOption = this._getNewOption(newData);
3534         $existingOption.replaceWith($newOption);
3535         continue;
3536     }
3537 }
3538
3539 var $option = this._getNewOption(item);
3540
3541 if (item.children) {
3542     var $children = this._convertToOptions(item.children);
3543     $option.appendMany($children);
3544 }
3545
3546 $options.push($option);
3547 }
3548
3549 return $options;
3550 };
3551
3552 return ArrayAdapter;
3553 };
3554
3555 // Define the select2 data adapter
3556 S2.define('select2/data/ajax',[
3557     './array',
3558     './utils',
3559     'jquery'
3560 ], function(ArrayAdapter, Utils, $) {
3561     function AjaxAdapter($element, options) {
3562         this.ajaxOptions = this._applyDefaults($.extend({}, options, {
3563             ajax: {}
3564         }));
3565
3566         if (this.ajaxOptions.processResults != null) {
3567             this.processResults = this.ajaxOptions.processResults;
3568         }
3569
3570         AjaxAdapter.__super__.constructor.call(this, $element, options);
3571     }
3572
3573     Utils.Extend(AjaxAdapter, ArrayAdapter);
3574
3575     AjaxAdapter.prototype._applyDefaults = function(options) {
3576         var defaults = {
3577             data: function(params) {
3578                 return $.extend({}, params, {
3579                     q: params.term
3580                 });
3581             },
3582             transport: function(params, success, failure) {
3583                 var $request = $.ajax(params);
3584
3585                 $request.then(success);
3586                 $request.fail(failure);
3587
3588                 return $request;
3589             }
3590         };
3591
3592         return $.extend({}, defaults, options, true);
3593     };
3594
3595     AjaxAdapter.prototype.processResults = function(results) {
3596         return results;
3597     };
3598
3599     AjaxAdapter.prototype.query = function(params, callback) {
3600         var matches = [];
3601         var self = this;
3602
3603         if (this._request != null) {
3604             // JSONP requests cannot always be aborted
3605             if (!$.isFunction(this._request.abort)) {
3606                 this._request.abort();
3607             }
3608
3609             this._request = null;
3610         }
3611
3612         var options = $.extend({
3613             type: 'GET'
3614         }, this.ajaxOptions);
3615
3616         if (typeof options.url == 'function') {
3617             options.url = options.url.call(this, $element, params);
3618         }
3619
3620         if (typeof options.data == 'function') {
3621             options.data = options.data.call(this, $element, params);
3622         }
3623
3624         function request() {
3625             var $request = options.transport(options, function(data) {
3626                 var results = self.processResults(data, params);
3627
3628                 if (self.options.get('debug') && window.console && console.error) {
3629                     // Check to make sure that the response included a `results` key.
3630                     if (!results || !$.isArray(results.results)) {
3631                         console.error(
3632                             'Select2: The AJAX results did not return an array in the ' +
3633                             '`results` key of the response.'
3634                         );
3635                     }
3636                 }
3637             });
3638         }
3639     };
3640
3641     return AjaxAdapter;
3642 }

```

```

3636     callback: results;
3637 }, function () {
3638     // Attempt to detect if a request was aborted
3639     // Only works if the transport exposes a status property
3640     if ('status' in $request &&
3641         $request.status === 0 || $request.status === '0') {
3642         return;
3643     }
3644
3645     self.trigger('results:message', {
3646         message: 'errorLoading'
3647     });
3648 });
3649
3650 self._request = $request;
3651 }
3652
3653 if (this.ajaxOptions.delay && params.term != null) {
3654     if (this._queryTimeout) {
3655         window.clearTimeout(this._queryTimeout);
3656     }
3657
3658     this._queryTimeout = window.setTimeout(request, this.ajaxOptions.delay);
3659 } else {
3660     request();
3661 }
3662
3663 };
3664
3665 return AjaxAdapter;
3666 });
3667
3668 S2.define('select2/data/tags',[
3669     'jquery'
3670 ], function ($) {
3671     function Tags(decorated $element, options) {
3672         var tags = options.get('tags');
3673
3674         var createTag = options.get('createTag');
3675
3676         if (createTag != undefined) {
3677             this.createTag = createTag;
3678         }
3679
3680         var insertTag = options.get('insertTag');
3681
3682         if (insertTag != undefined) {
3683             this.insertTag = insertTag;
3684         }
3685
3686         decorated.call(this, $element, options);
3687
3688         if ($.isArray(tags)) {
3689             for (var t = 0; t < tags.length; t++) {
3690                 var tag = tags[t];
3691                 var item = this._normalizeItem(tag);
3692
3693                 var $option = this._option(item);
3694
3695                 this.$element.append($option);
3696             }
3697         }
3698     }
3699
3700     Tags.prototype.query = function(decorated, params, callback) {
3701         var self = this;
3702
3703         this._removeOldTags();
3704
3705         if (params.term == null || params.page != null) {
3706             decorated.call(this, params, callback);
3707             return;
3708         }
3709
3710         function wrapper(obj, child) {
3711             var data = obj.results;
3712
3713             for (var i = 0; i < data.length; i++) {
3714                 var option = data[i];
3715
3716                 var checkChildren = (
3717                     option.children != null &&
3718                     !wrapper({
3719                         results: option.children
3720                     }, true)
3721                 );
3722
3723                 var optionText = (option.text || '').toUpperCase();
3724                 var paramsTerm = (params.term || '').toUpperCase();
3725
3726                 var checkText = optionText === paramsTerm;
3727
3728                 if (checkText || checkChildren) {
3729                     if (child) {
3730                         return false;
3731                     }
3732
3733                     obj.data = data;
3734                     callback(obj);
3735
3736                     return;
3737                 }
3738             }
3739
3740             if (child) {
3741                 return true;
3742             }
3743
3744             var tag = self.createTag(params);
3745
3746             if (tag != null) {
3747                 var $option = self._option(tag);
3748                 $option.attr('data-select2-tag', true);
3749
3750                 self.addOptions([$option]);

```

```

3751     self.insertTag(data, tag);
3752 }
3753
3754 obj.results = data;
3755
3756 callback(obj);
3757 }
3758
3759 decorated.call(this, params, wrapper);
3760 };
3761
3762 Tags.prototype.createTag = function (decorated, params) {
3763     var term = $.trim(params.term);
3764
3765     if (term === "") {
3766         return null;
3767     }
3768
3769     return {
3770         id: term,
3771         text: term
3772     };
3773 };
3774
3775 Tags.prototype.insertTag = function (_, data, tag) {
3776     data.unshift(tag);
3777 };
3778
3779 Tags.prototype._removeOldTags = function (_) {
3780     var $options = this.$element.find("option[data-select2-tag]");
3781
3782     $options.each(function () {
3783         if (this.selected) {
3784             return;
3785         }
3786     });
3787
3788     $(this).remove();
3789 };
3790
3791 return Tags;
3792
3793 },
3794
3795 S2.define('select2/data/tokenizer',[
3796     'jquery'
3797 ], function ($) {
3798     function Tokenizer(decorated, $element, options) {
3799         var tokenizer = options.get('tokenizer');
3800
3801         if (tokenizer !== undefined) {
3802             this.tokenizer = tokenizer;
3803         }
3804
3805         decorated.call(this, $element, options);
3806     }
3807
3808     Tokenizer.prototype.bind = function (decorated, container, $container) {
3809         decorated.call(this, container, $container);
3810
3811         this.$search = container.dropdown.$search || container.selection.$search ||
3812             $container.find('.select2-search__field');
3813     };
3814
3815     Tokenizer.prototype.query = function (decorated, params, callback) {
3816         var self = this;
3817
3818         function createAndSelect (data) {
3819             // Normalize the data object so we can use it for checks
3820             var item = self._normalizeItem(data);
3821
3822             // Check if the data object already exists as a tag
3823             // Select it if it doesn't
3824             var $existingOptions = self.$element.find('option').filter(function () {
3825                 return $(this).val() === item.id;
3826             });
3827
3828             // If an existing option wasn't found for it, create the option
3829             if (!$existingOptions.length) {
3830                 var $option = self._option(item);
3831                 $option.attr('data-select2-tag', true);
3832
3833                 self._removeOldTags();
3834                 self.addOptions([$option]);
3835             }
3836
3837             // Select the item, now that we know there is an option for it
3838             select(item);
3839         }
3840
3841         function select (data) {
3842             self.trigger('select', {
3843                 data: data
3844             });
3845         }
3846
3847         params.term = params.term || "";
3848
3849         var tokenData = this.tokenizer(params, this.options, createAndSelect);
3850
3851         if (tokenData.term !== params.term) {
3852             // Replace the search term if we have the search box
3853             if (this.$search.length) {
3854                 this.$search.val(tokenData.term);
3855                 this.$search.trigger('focus');
3856             }
3857
3858             params.term = tokenData.term;
3859         }
3860
3861         decorated.call(this, params, callback);
3862     };
3863
3864     Tokenizer.prototype.tokenizer = function (_, params, options, callback) {
3865         var separators = options.get('tokenSeparators') || [];

```

```

3866 var term = params.term;
3867 var i = 0;
3868
3869 var createTag = this.createTag || function (params) {
3870   return {
3871     id: params.term,
3872     text: params.term
3873   };
3874 };
3875
3876 while (i < term.length) {
3877   var termChar = term[i];
3878
3879   if ($.inArray(termChar, separators) === -1) {
3880     i++;
3881     continue;
3882   }
3883
3884   var part = term.substr(0, i);
3885   var partParams = $.extend({}, params, {
3886     term: part
3887   });
3888
3889   var data = createTag(partParams);
3890
3891   if (data == null) {
3892     i++;
3893     continue;
3894   }
3895
3896   callback(data);
3897
3898   // Reset the term to not include the tokenized portion
3899   term = term.substr(i + 1) || "";
3900   i = 0;
3901 }
3902
3903 return {
3904   term: term
3905 };
3906 };
3907 };
3908
3909 return Tokenizer;
3910 });
3911
3912 S2.define('select2/data/minimumInputLength',[
3913 ],function () {
3914   function MinimumInputLength (decorated, $e, options) {
3915     this.minimumInputLength = options.get('minimumInputLength');
3916
3917     decorated.call(this, $e, options);
3918   }
3919
3920   MinimumInputLength.prototype.query = function (decorated, params, callback) {
3921     params.term = params.term || "";
3922
3923     if (params.term.length < this.minimumInputLength) {
3924       this.trigger('results:message', {
3925         message: 'inputTooShort',
3926         args: {
3927           minimum: this.minimumInputLength,
3928           input: params.term,
3929           params: params
3930         }
3931       });
3932     }
3933
3934     return;
3935   }
3936
3937   decorated.call(this, params, callback);
3938 };
3939
3940 return MinimumInputLength;
3941 });
3942
3943 S2.define('select2/data/maximumInputLength',[
3944 ],function () {
3945   function MaximumInputLength (decorated, $e, options) {
3946     this.maximumInputLength = options.get('maximumInputLength');
3947
3948     decorated.call(this, $e, options);
3949   }
3950
3951   MaximumInputLength.prototype.query = function (decorated, params, callback) {
3952     params.term = params.term || "";
3953
3954     if (this.maximumInputLength > 0 &&
3955         params.term.length > this.maximumInputLength) {
3956       this.trigger('results:message', {
3957         message: 'inputTooLong',
3958         args: {
3959           maximum: this.maximumInputLength,
3960           input: params.term,
3961           params: params
3962         }
3963       });
3964     }
3965
3966     return;
3967   }
3968
3969   decorated.call(this, params, callback);
3970 };
3971
3972 return MaximumInputLength;
3973 });
3974
3975 S2.define('select2/data/maximumSelectionLength',[
3976 ],function () {
3977   function MaximumSelectionLength (decorated, $e, options) {
3978     this.maximumSelectionLength = options.get('maximumSelectionLength');
3979
3980

```

```

3981     decorated call(this, $e, options);
3982 }
3983
3984 MaximumSelectionLength.prototype bind =
3985     function (decorated, container, $container) {
3986         var self = this;
3987
3988         decorated call(this, container, $container);
3989
3990         container.on('select', function () {
3991             self._checkIfMaximumSelected();
3992         });
3993     };
3994
3995 MaximumSelectionLength.prototype query =
3996     function (decorated, params, callback) {
3997         var self = this;
3998
3999         this._checkIfMaximumSelected(function () {
4000             decorated call(self, params, callback);
4001         });
4002     };
4003
4004 MaximumSelectionLength.prototype._checkIfMaximumSelected =
4005     function (_, successCallback) {
4006         var self = this;
4007
4008         this.current(function (currentData) {
4009             var count = currentData != null ? currentData.length : 0;
4010             if (self.maximumSelectionLength > 0 &&
4011                 count >= self.maximumSelectionLength) {
4012                 self.trigger('results:message', {
4013                     message: 'maximumSelected',
4014                     args: {
4015                         maximum: self.maximumSelectionLength
4016                     }
4017                 });
4018                 return;
4019             }
4020
4021             if (successCallback) {
4022                 successCallback();
4023             }
4024         });
4025     };
4026
4027     return MaximumSelectionLength;
4028 });
4029
4030 S2.define('select2/dropdown',[
4031     'jquery',
4032     './utils'
4033 ], function ($, Utils) {
4034     function Dropdown ($element, options) {
4035         this.$element = $element;
4036         this.options = options;
4037
4038         Dropdown.__super__.constructor.call(this);
4039     }
4040
4041     Utils.Extend(Dropdown, Utils.Observable);
4042
4043     Dropdown.prototype.render = function () {
4044         var $dropdown = $(
4045             '<span class="select2-dropdown">' +
4046             '<span class="select2-results"></span>' +
4047             '</span>'
4048         );
4049
4050         $dropdown.attr('dir', this.options.get('dir'));
4051
4052         this.$dropdown = $dropdown;
4053
4054         return $dropdown;
4055     };
4056
4057     Dropdown.prototype.bind = function () {
4058         // Should be implemented in subclasses
4059     };
4060
4061     Dropdown.prototype.position = function ($dropdown, $container) {
4062         // Should be implemented in subclasses
4063     };
4064
4065     Dropdown.prototype.destroy = function () {
4066         // Remove the dropdown from the DOM
4067         this.$dropdown.remove();
4068     };
4069
4070     return Dropdown;
4071 });
4072
4073 S2.define('select2/dropdown/search',[
4074     'jquery',
4075     './utils'
4076 ], function ($, Utils) {
4077     function Search () { }
4078
4079     Search.prototype.render = function (decorated) {
4080         var $rendered = decorated.call(this);
4081
4082         var $search = $(
4083             '<span class="select2-search select2-search--dropdown">' +
4084             '<input class="select2-search__field" type="search" tabindex="-1" +
4085             ' autocomplete="off" autocorrect="off" autocapitalize="none" +
4086             ' spellcheck="false" role="searchbox" aria-autocomplete="list" />' +
4087             '</span>'
4088         );
4089
4090         this.$searchContainer = $search;
4091         this.$search = $search.find('input');
4092
4093         $rendered.prepend($search);
4094
4095         return $rendered;

```

```

4096 };
4097
4098 Search.prototype bind = function (decorated container, $container) {
4099     var self = this;
4100
4101     var resultsId = container.id + '-results';
4102
4103     decorated call(this, container, $container);
4104
4105     this.$search on('keydown', function (evt) {
4106         self trigger('keypress', evt);
4107
4108         self._keyUpPrevented = evt.isDefaultPrevented();
4109     });
4110
4111     // Workaround for browsers which do not support the `input` event
4112     // This will prevent double-triggering of events for browsers which support
4113     // both the `keyup` and `input` events.
4114     this.$search on('input', function (evt) {
4115         // Unbind the duplicated `keyup` event
4116         $(this).off('keyup');
4117     });
4118
4119     this.$search on('keyup input', function (evt) {
4120         self handleSearch evt;
4121     });
4122
4123     container on('open', function () {
4124         self.$search attr('tabindex', 0);
4125         self.$search attr('aria-controls', resultsId);
4126
4127         self.$search trigger('focus');
4128
4129         window.setTimeout(function () {
4130             self.$search trigger('focus');
4131         }, 0);
4132     });
4133
4134     container on('close', function () {
4135         self.$search attr('tabindex', -1);
4136         self.$search removeAttr('aria-controls');
4137         self.$search removeAttr('aria-activedescendant');
4138
4139         self.$search val('');
4140         self.$search trigger('blur');
4141     });
4142
4143     container on('focus', function () {
4144         if (!container.isOpen()) {
4145             self.$search trigger('focus');
4146         }
4147     });
4148
4149     container on('results:all', function (params) {
4150         if (params.query.term == null || params.query.term === '') {
4151             var showSearch = self.showSearch(params);
4152
4153             if (showSearch) {
4154                 self.$searchContainer.removeClass('select2-search--hide');
4155             } else {
4156                 self.$searchContainer.addClass('select2-search--hide');
4157             }
4158         }
4159     });
4160
4161     container on('results:focus', function (params) {
4162         if (params.data._resultId) {
4163             self.$search attr('aria-activedescendant', params.data._resultId);
4164         } else {
4165             self.$search removeAttr('aria-activedescendant');
4166         }
4167     });
4168 };
4169
4170 Search.prototype handleSearch = function (evt) {
4171     if (!this._keyUpPrevented) {
4172         var input = this.$search.val();
4173
4174         this.trigger('query', {
4175             term: input
4176         });
4177     }
4178
4179     this._keyUpPrevented = false;
4180 };
4181
4182 Search.prototype showSearch = function (_, params) {
4183     return true;
4184 };
4185
4186 Search.prototype showSearch = function (params) {
4187     return true;
4188 };
4189
4190 S2.define('select2/dropdown/hidePlaceholder',[
4191 ],function () {
4192     function HidePlaceholder (decorated $element, options, dataAdapter) {
4193         this.placeholder = this.normalizePlaceholder(options.get('placeholder'));
4194     }
4195
4196     decorated call(this, $element, options, dataAdapter);
4197
4198     HidePlaceholder.prototype append = function (decorated data) {
4199         data.results = this.removePlaceholder(data.results);
4200
4201         decorated call(this, data);
4202     };
4203
4204     HidePlaceholder.prototype normalizePlaceholder = function (_, placeholder) {
4205         if (typeof placeholder == 'string') {
4206             placeholder = {
4207                 id: "",
4208                 text: placeholder
4209             };
4210         }
4211     };

```

```

4211     return placeholder;
4212 };
4213
4214 HidePlaceholder.prototype.removePlaceholder = function (_ data) {
4215     var modifiedData = data.slice(0);
4216     for (var d = data.length - 1; d >= 0; d--) {
4217         var item = data[d];
4218         if (this.placeholder.id === item.id) {
4219             modifiedData.splice(d, 1);
4220         }
4221     }
4222     return modifiedData;
4223 };
4224
4225 return HidePlaceholder;
4226
4227
4228
4229 return HidePlaceholder;
4230 });
4231
4232 S2.define('select2/dropdown/infiniteScroll',[
4233     'jquery'
4234 ], function ($) {
4235     function InfiniteScroll (decorated, $element, options, dataAdapter) {
4236         this.lastParams = {};
4237         decorated.call(this, $element, options, dataAdapter);
4238         this.$loadingMore = this.createLoadingMore();
4239         this.loading = false;
4240     }
4241
4242     InfiniteScroll.prototype.append = function (decorated, data) {
4243         this.$loadingMore.remove();
4244         this.loading = false;
4245         decorated.call(this, data);
4246         if (this.showLoadingMore(data)) {
4247             this.$results.append(this.$loadingMore);
4248             this.loadMoreIfNeeded();
4249         }
4250     };
4251
4252     InfiniteScroll.prototype.bind = function (decorated, container, $container) {
4253         var self = this;
4254         decorated.call(this, container, $container);
4255         container.on('query', function (params) {
4256             self.lastParams = params;
4257             self.loading = true;
4258         });
4259         container.on('query:append', function (params) {
4260             self.lastParams = params;
4261             self.loading = true;
4262         });
4263         this.$results.on('scroll', this.loadMoreIfNeeded.bind(this));
4264     };
4265
4266     InfiniteScroll.prototype.loadMoreIfNeeded = function () {
4267         var isLoadMoreVisible = $.contains(
4268             document.documentElement,
4269             this.$loadingMore[0]
4270         );
4271         if (this.loading || !isLoadMoreVisible) {
4272             return;
4273         }
4274         var currentOffset = this.$results.offset().top +
4275             this.$results.outerHeight(false);
4276         var loadingMoreOffset = this.$loadingMore.offset().top +
4277             this.$loadingMore.outerHeight(false);
4278         if (currentOffset + 50 >= loadingMoreOffset) {
4279             this.loadMore();
4280         }
4281     };
4282
4283     InfiniteScroll.prototype.loadMore = function () {
4284         this.loading = true;
4285         var params = $.extend({}, this.lastParams);
4286         params.page++;
4287         this.trigger('query:append', params);
4288     };
4289
4290     InfiniteScroll.prototype.showLoadingMore = function (_ data) {
4291         return data.pagination && data.pagination.more;
4292     };
4293
4294     InfiniteScroll.prototype.createLoadingMore = function () {
4295         var $option = $(
4296             '<li> +
4297             'class="select2-results__option select2-results__option--load-more"' +
4298             'role="option" aria-disabled="true"></li>'
4299         );
4300         var message = this.options.get('translations').get('loadingMore');
4301         $option.html(message(this.lastParams));
4302         return $option;
4303     };
4304
4305     return InfiniteScroll;
4306 });
4307
4308 S2.define('select2/dropdown/attachBody',[

```



```

4326 'jquery',
4327 './utils'
4328 }, function ($, Utils) {
4329     function AttachBody (decorated, $element, options) {
4330         this.$dropdownParent = $.options.get('dropdownParent') || document body;
4331     }
4332     decorated call(this, $element, options);
4333 }
4334
4335 AttachBody prototype bind = function (decorated, container, $container) {
4336     var self = this;
4337     decorated call(this, container, $container);
4338
4339     container on('open', function () {
4340         self._showDropdown();
4341         self._attachPositioningHandler(container);
4342     });
4343     // Must bind after the results handlers to ensure correct sizing
4344     self._bindContainerResultHandlers(container);
4345 };
4346
4347 container on('close', function () {
4348     self._hideDropdown();
4349     self._detachPositioningHandler(container);
4350 });
4351
4352 this.$dropdownContainer on('mousedown', function (evt) {
4353     evt.stopPropagation();
4354 });
4355
4356 AttachBody prototype destroy = function (decorated) {
4357     decorated call(this);
4358     this.$dropdownContainer remove();
4359 };
4360
4361 AttachBody prototype position = function (decorated, $dropdown, $container) {
4362     // Clone all of the container classes
4363     $dropdown attr('class', $container attr('class'));
4364     $dropdown removeClass('select2');
4365     $dropdown addClass('select2-container--open');
4366
4367     $dropdown css({
4368         position: 'absolute',
4369         top: '-99999px'
4370     });
4371
4372     this.$container = $container;
4373 };
4374
4375 AttachBody prototype render = function (decorated) {
4376     var $container = $('<span></span>');
4377     var $dropdown = decorated call(this);
4378     $container append($dropdown);
4379     this.$dropdownContainer = $container;
4380     return $container;
4381 };
4382
4383 AttachBody prototype _hideDropdown = function (decorated) {
4384     this.$dropdownContainer detach();
4385 };
4386
4387 AttachBody prototype _bindContainerResultHandlers =
4388     function (decorated, container) {
4389         // These should only be bound once
4390         if (this._containerResultsHandlersBound) {
4391             return;
4392         }
4393         var self = this;
4394         container on('results:all', function () {
4395             self._positionDropdown();
4396             self._resizeDropdown();
4397         });
4398         container on('results:append', function () {
4399             self._positionDropdown();
4400             self._resizeDropdown();
4401         });
4402         container on('results:message', function () {
4403             self._positionDropdown();
4404             self._resizeDropdown();
4405         });
4406         container on('select', function () {
4407             self._positionDropdown();
4408             self._resizeDropdown();
4409         });
4410         container on('unselect', function () {
4411             self._positionDropdown();
4412             self._resizeDropdown();
4413         });
4414         this._containerResultsHandlersBound = true;
4415     };
4416
4417 AttachBody prototype _attachPositioningHandler =
4418     function (decorated, container) {
4419         var self = this;
4420         var scrollEvent = 'scroll.select2.' + container id;
4421         var resizeEvent = 'resize.select2.' + container id;
4422         var orientationEvent = 'orientationchange.select2.' + container id;
4423         var $swatchers = this.$container parents().filter(Utils.hasScroll);

```

```

4441 $watchers each(function () {
4442   Utils.StoreData(this, 'select2-scroll-position', {
4443     x: $(this).scrollLeft(),
4444     y: $(this).scrollTop()
4445   });
4446 });
4447
4448 $watchers.on('scrollEvent', function (ev) {
4449   var position = Utils.GetData(this, 'select2-scroll-position');
4450   $(this).scrollTop(position.y);
4451 });
4452
4453 $(window).on('scrollEvent' + ' ' + 'resizeEvent' + ' ' + 'orientationEvent',
4454   function (e) {
4455     self._positionDropdown();
4456     self._resizeDropdown();
4457   });
4458 });
4459
4460 AttachBody.prototype._detachPositioningHandler =
4461   function (decorated, container) {
4462     var scrollEvent = 'scroll.select2.' + container.id;
4463     var resizeEvent = 'resize.select2.' + container.id;
4464     var orientationEvent = 'orientationchange.select2.' + container.id;
4465
4466     var $watchers = this.$container.parents().filter(Utils.hasScroll);
4467     $watchers.off(scrollEvent);
4468
4469     $(window).off('scrollEvent' + ' ' + 'resizeEvent' + ' ' + 'orientationEvent');
4470   };
4471
4472 AttachBody.prototype._positionDropdown = function () {
4473   var $window = $(window);
4474
4475   var isCurrentlyAbove = this.$dropdown.hasClass('select2-dropdown--above');
4476   var isCurrentlyBelow = this.$dropdown.hasClass('select2-dropdown--below');
4477
4478   var newDirection = null;
4479
4480   var offset = this.$container.offset();
4481
4482   offset.bottom = offset.top + this.$container.outerHeight(false);
4483
4484   var container = {
4485     height: this.$container.outerHeight(false)
4486   };
4487
4488   container.top = offset.top;
4489   container.bottom = offset.top + container.height;
4490
4491   var dropdown = {
4492     height: this.$dropdown.outerHeight(false)
4493   };
4494
4495   var viewport = {
4496     top: $window.scrollTop(),
4497     bottom: $window.scrollTop() + $window.height()
4498   };
4499
4500   var enoughRoomAbove = viewport.top < (offset.top - dropdown.height);
4501   var enoughRoomBelow = viewport.bottom > (offset.bottom + dropdown.height);
4502
4503   var css = {
4504     left: offset.left,
4505     top: container.bottom
4506   };
4507
4508   // Determine what the parent element is to use for calculating the offset
4509   var $offsetParent = this.$dropdownParent;
4510
4511   // For statically positioned elements, we need to get the element
4512   // that is determining the offset
4513   if ($offsetParent.css('position') === 'static') {
4514     $offsetParent = $offsetParent.offsetParent();
4515   }
4516
4517   var parentOffset = {
4518     top: 0,
4519     left: 0
4520   };
4521
4522   if (
4523     $contains(document.body, $offsetParent[0]) ||
4524     $offsetParent[0].isConnected
4525   ) {
4526     parentOffset = $offsetParent.offset();
4527   }
4528
4529   css.top -= parentOffset.top;
4530   css.left -= parentOffset.left;
4531
4532   if (!isCurrentlyAbove && !isCurrentlyBelow) {
4533     newDirection = 'below';
4534   }
4535
4536   if (!enoughRoomBelow && enoughRoomAbove && !isCurrentlyAbove) {
4537     newDirection = 'above';
4538   } else if (!enoughRoomAbove && enoughRoomBelow && !isCurrentlyBelow) {
4539     newDirection = 'below';
4540   }
4541
4542   if (newDirection === 'above') {
4543     (isCurrentlyAbove && newDirection !== 'below') ?
4544     css.top = container.top - parentOffset.top - dropdown.height :
4545     ;
4546   }
4547
4548   if (newDirection !== null) {
4549     this.$dropdown
4550       .removeClass('select2-dropdown--below select2-dropdown--above')
4551       .addClass('select2-dropdown--' + newDirection);
4552     this.$container
4553       .removeClass('select2-container--below select2-container--above')
4554       .addClass('select2-container--' + newDirection);
4555   }

```

```

4556     this.$dropdownContainer.css(css);
4557 };
4558
4559 AttachBody.prototype._resizeDropdown = function () {
4560     var css = {
4561         width: this.$container.outerWidth(false) + 'px'
4562     };
4563
4564     if (this.options.get('dropdownAutoWidth')) {
4565         css.minWidth = css.width;
4566         css.position = 'relative';
4567         css.width = 'auto';
4568     }
4569
4570     this.$dropdown.css(css);
4571 };
4572
4573 AttachBody.prototype._showDropdown = function (decorated) {
4574     this.$dropdownContainer.appendTo(this.$dropdownParent);
4575
4576     this._positionDropdown();
4577     this._resizeDropdown();
4578 };
4579
4580 return AttachBody;
4581 });
4582
4583 S2.define('select2/dropdown/minimumResultsForSearch',[
4584     function () {
4585         function countResults (data) {
4586             var count = 0;
4587
4588             for (var d = 0; d < data.length; d++) {
4589                 var item = data[d];
4590
4591                 if (item.children) {
4592                     count += countResults(item.children);
4593                 } else {
4594                     count++;
4595                 }
4596             }
4597
4598             return count;
4599         }
4600     },
4601     function (decorated, $element, options, dataAdapter) {
4602         this.minimumResultsForSearch = options.get('minimumResultsForSearch');
4603
4604         if (this.minimumResultsForSearch < 0) {
4605             this.minimumResultsForSearch = Infinity;
4606         }
4607
4608         decorated.call(this, $element, options, dataAdapter);
4609     },
4610     function (decorated, params) {
4611         if (countResults(params.data.results) < this.minimumResultsForSearch) {
4612             return false;
4613         }
4614
4615         return decorated.call(this, params);
4616     }
4617 );
4618
4619 return MinimumResultsForSearch;
4620 });
4621
4622 S2.define('select2/dropdown/selectOnClose',[
4623     './utils'
4624 ], function (Utils) {
4625     function SelectOnClose () {}
4626
4627     SelectOnClose.prototype.bind = function (decorated, container, $container) {
4628         var self = this;
4629
4630         decorated.call(this, container, $container);
4631
4632         container.on('close', function (params) {
4633             self._handleSelectOnClose(params);
4634         });
4635     };
4636
4637     SelectOnClose.prototype._handleSelectOnClose = function (_ params) {
4638         if (params.originalSelect2Event != null) {
4639             var event = params.originalSelect2Event;
4640
4641             // Don't select an item if the close event was triggered from a select or
4642             // unselect event
4643             if (event.type === 'select' || event.type === 'unselect') {
4644                 return;
4645             }
4646
4647             var $highlightedResults = this.getHighlightedResults();
4648
4649             // Only select highlighted results
4650             if ($highlightedResults.length < 1) {
4651                 return;
4652             }
4653
4654             var data = Utils.GetData($highlightedResults[0], 'data');
4655
4656             // Don't re-select already selected results
4657             if (
4658                 (data.element != null && data.element.selected) ||
4659                 (data.element == null && data.selected)
4660             ) {
4661                 return;
4662             }
4663
4664             this.trigger('select', {
4665                 data: data
4666             });
4667         }
4668     };
4669
4670

```

```

4671 return SelectOnClose;
4672 });
4673
4674 S2.define('select2/dropdown/closeOnSelect',[
4675
4676 ],function () {
4677     function CloseOnSelect () { }
4678
4679     CloseOnSelect.prototype.bind = function (decorated, container, $container) {
4680         var self = this;
4681
4682         decorated.call(this, container, $container);
4683
4684         container.on('select', function (evt) {
4685             self._selectTriggered(evt);
4686         });
4687
4688         container.on('unselect', function (evt) {
4689             self._selectTriggered(evt);
4690         });
4691     };
4692
4693     CloseOnSelect.prototype._selectTriggered = function (_, evt) {
4694         var originalEvent = evt.originalEvent;
4695
4696         // Don't close if the control key is being held
4697         if (originalEvent && (originalEvent.ctrlKey || originalEvent.metaKey)) {
4698             return;
4699         }
4700
4701         this.trigger('close', {
4702             originalEvent: originalEvent,
4703             originalSelect2Event: evt
4704         });
4705     };
4706
4707     return CloseOnSelect;
4708 });
4709
4710 S2.define('select2/i18n/en',[],function () {
4711     // English
4712     return {
4713         errorLoading: function () {
4714             return 'The results could not be loaded.';
4715         },
4716         inputTooLong: function (args) {
4717             var overChars = args.input.length - args.maximum;
4718
4719             var message = 'Please delete ' + overChars + ' character';
4720
4721             if (overChars != 1) {
4722                 message += 's';
4723             }
4724
4725             return message;
4726         },
4727         inputTooShort: function (args) {
4728             var remainingChars = args.minimum - args.input.length;
4729
4730             var message = 'Please enter ' + remainingChars + ' or more characters';
4731
4732             return message;
4733         },
4734         loadingMore: function () {
4735             return 'Loading more results';
4736         },
4737         maximumSelected: function (args) {
4738             var message = 'You can only select ' + args.maximum + ' item';
4739
4740             if (args.maximum != 1) {
4741                 message += 's';
4742             }
4743
4744             return message;
4745         },
4746         noResults: function () {
4747             return 'No results found';
4748         },
4749         searching: function () {
4750             return 'Searching';
4751         },
4752         removeAllItems: function () {
4753             return 'Remove all items';
4754         }
4755     };
4756 });
4757
4758 S2.define('select2/defaults',[
4759     'jquery',
4760     'require',
4761
4762     './results',
4763
4764     './selection/single',
4765     './selection/multiple',
4766     './selection/placeholder',
4767     './selection/allowClear',
4768     './selection/search',
4769     './selection/eventRelay',
4770
4771     './utils',
4772     './translation',
4773     './diacritics',
4774
4775     './data/select',
4776     './data/array',
4777     './data/ajax',
4778     './data/tags',
4779     './data/tokenizer',
4780     './data/minimumInputLength',
4781     './data/maximumInputLength',
4782     './data/maximumSelectionLength',
4783
4784     './dropdown',
4785     './dropdown/search',

```

```

4786 './dropdown/hidePlaceholder',
4787 './dropdown/infiniteScroll',
4788 './dropdown/attachBody',
4789 './dropdown/minimumResultsForSearch',
4790 './dropdown/selectOnClose',
4791 './dropdown/closeOnSelect',
4792
4793 './!18n/en'
4794 ], function ($) {
4795     require(
4796         ResultsList,
4797         SingleSelection, MultipleSelection, Placeholder, AllowClear,
4798         SelectionSearch, EventRelay,
4799
4800         Utils, Translation, DIACRITICS,
4801
4802         SelectData, ArrayData, AjaxData, Tags, Tokenizer,
4803         MinimumInputLength, MaximumInputLength, MaximumSelectionLength,
4804
4805         Dropdown, DropdownSearch, HidePlaceholder, InfiniteScroll,
4806         AttachBody, MinimumResultsForSearch, SelectOnClose, CloseOnSelect,
4807
4808         EnglishTranslation) {
4809     function Defaults () {
4810     this.reset();
4811     }
4812
4813     Defaults.prototype.apply = function (options) {
4814     options = $.extend(true, {}, this.defaults, options);
4815
4816     if (options.dataAdapter == null) {
4817     if (options.ajax != null) {
4818     options.dataAdapter = AjaxData;
4819     } else if (options.data != null) {
4820     options.dataAdapter = ArrayData;
4821     } else {
4822     options.dataAdapter = SelectData;
4823     }
4824
4825     if (options.minimumInputLength > 0) {
4826     options.dataAdapter = Utils.Decorate(
4827     options.dataAdapter,
4828     MinimumInputLength
4829     );
4830
4831     if (options.maximumInputLength > 0) {
4832     options.dataAdapter = Utils.Decorate(
4833     options.dataAdapter,
4834     MaximumInputLength
4835     );
4836
4837     if (options.maximumSelectionLength > 0) {
4838     options.dataAdapter = Utils.Decorate(
4839     options.dataAdapter,
4840     MaximumSelectionLength
4841     );
4842
4843     if (options.tags) {
4844     options.dataAdapter = Utils.Decorate(options.dataAdapter, Tags);
4845     }
4846
4847     if (options.tokenSeparators != null || options.tokenizer != null) {
4848     options.dataAdapter = Utils.Decorate(
4849     options.dataAdapter,
4850     Tokenizer
4851     );
4852
4853     if (options.query != null) {
4854     var Query = require(options.amdBase + 'compat/query');
4855
4856     options.dataAdapter = Utils.Decorate(
4857     options.dataAdapter,
4858     Query
4859     );
4860
4861     if (options.initSelection != null) {
4862     var InitSelection = require(options.amdBase + 'compat/initSelection');
4863
4864     options.dataAdapter = Utils.Decorate(
4865     options.dataAdapter,
4866     InitSelection
4867     );
4868
4869     if (options.resultsAdapter == null) {
4870     options.resultsAdapter = ResultsList;
4871
4872     if (options.ajax != null) {
4873     options.resultsAdapter = Utils.Decorate(
4874     options.resultsAdapter,
4875     InfiniteScroll
4876     );
4877
4878     if (options.placeholder != null) {
4879     options.resultsAdapter = Utils.Decorate(
4880     options.resultsAdapter,
4881     HidePlaceholder
4882     );
4883
4884     if (options.selectOnClose) {
4885     options.resultsAdapter = Utils.Decorate(
4886     options.resultsAdapter,
4887     SelectOnClose
4888     );
4889
4890     }

```

```

4901
4902 if (options dropdownAdapter == null) {
4903   if (options multiple) {
4904     options dropdownAdapter = Dropdown;
4905   } else {
4906     var SearchableDropdown = Utils Decorate(Dropdown, DropdownSearch);
4907
4908     options dropdownAdapter = SearchableDropdown;
4909   }
4910
4911   if (options minimumResultsForSearch != 0) {
4912     options dropdownAdapter = Utils Decorate(
4913       options dropdownAdapter,
4914       MinimumResultsForSearch
4915     );
4916   }
4917
4918   if (options closeOnSelect) {
4919     options dropdownAdapter = Utils Decorate(
4920       options dropdownAdapter,
4921       CloseOnSelect
4922     );
4923   }
4924
4925   if {
4926     options dropdownCssClass != null ||
4927     options dropdownCss != null ||
4928     options adaptDropdownCssClass != null
4929   } {
4930     var DropdownCSS = require(options amdBase + 'compat/dropdownCss');
4931
4932     options dropdownAdapter = Utils Decorate(
4933       options dropdownAdapter,
4934       DropdownCSS
4935     );
4936   }
4937
4938   options dropdownAdapter = Utils Decorate(
4939     options dropdownAdapter,
4940     AttachBody
4941   );
4942 }
4943
4944 if (options selectionAdapter == null) {
4945   if (options multiple) {
4946     options selectionAdapter = MultipleSelection;
4947   } else {
4948     options selectionAdapter = SingleSelection;
4949   }
4950
4951   // Add the placeholder mixin if a placeholder was specified
4952   if (options placeholder != null) {
4953     options selectionAdapter = Utils Decorate(
4954       options selectionAdapter,
4955       Placeholder
4956     );
4957   }
4958
4959   if (options allowClear) {
4960     options selectionAdapter = Utils Decorate(
4961       options selectionAdapter,
4962       AllowClear
4963     );
4964   }
4965
4966   if (options multiple) {
4967     options selectionAdapter = Utils Decorate(
4968       options selectionAdapter,
4969       SelectionSearch
4970     );
4971   }
4972
4973   if {
4974     options containerCssClass != null ||
4975     options containerCss != null ||
4976     options adaptContainerCssClass != null
4977   } {
4978     var ContainerCSS = require(options amdBase + 'compat/containerCss');
4979
4980     options selectionAdapter = Utils Decorate(
4981       options selectionAdapter,
4982       ContainerCSS
4983     );
4984   }
4985
4986   options selectionAdapter = Utils Decorate(
4987     options selectionAdapter,
4988     EventRelay
4989   );
4990 }
4991
4992 // If the defaults were not previously applied from an element, it is
4993 // possible for the language option to have not been resolved
4994 options language = this._resolveLanguage(options language);
4995
4996 // Always fall back to English since it will always be complete
4997 options language.push('en');
4998
4999 var uniqueLanguages = [];
5000
5001 for (var I = 0, l < options language.length; I++) {
5002   var language = options language[I];
5003
5004   if (uniqueLanguages.indexOf(language) === -1) {
5005     uniqueLanguages.push(language);
5006   }
5007 }
5008
5009 options language = uniqueLanguages;
5010
5011 options translations = this._processTranslations(
5012   options language
5013   options debug
5014 );
5015

```

```

5016     return options;
5017 };
5018
5019 Defaults.prototype.reset = function () {
5020     function stripDiacritics (text) {
5021         // Used 'uni range + named function' from http://jsperf.com/diacritics/18
5022         function match(a) {
5023             return DIACRITICS[a] || a;
5024         }
5025
5026         return text.replace(/[\u0000-\u007E]/g, match);
5027     }
5028
5029     function matcher (params, data) {
5030         // Always return the object if there is nothing to compare
5031         if (!$.trim(params.term) === "") {
5032             return data;
5033         }
5034
5035         // Do a recursive check for options with children
5036         if (data.children && data.children.length > 0) {
5037             // Clone the data object if there are children
5038             // This is required as we modify the object to remove any non-matches
5039             var match = $.extend(true, {}, data);
5040
5041             // Check each child of the option
5042             for (var c = data.children.length - 1; c >= 0; c--) {
5043                 var child = data.children[c];
5044
5045                 var matches = matcher(params, child);
5046
5047                 // If there wasn't a match, remove the object in the array
5048                 if (matches === null) {
5049                     match.children.splice(c, 1);
5050                 }
5051             }
5052
5053             // If any children matched, return the new object
5054             if (match.children.length > 0) {
5055                 return match;
5056             }
5057
5058             // If there were no matching children, check just the plain object
5059             return matcher(params, match);
5060         }
5061
5062         var original = stripDiacritics(data.text).toUpperCase();
5063         var term = stripDiacritics(params.term).toUpperCase();
5064
5065         // Check if the text contains the term
5066         if (original.indexOf(term) > -1) {
5067             return data;
5068         }
5069
5070         // If it doesn't contain the term, don't return anything
5071         return null;
5072     }
5073
5074     this.defaults = {
5075         amdBase: './',
5076         amdLanguageBase: './i18n/',
5077         closeOnSelect: true,
5078         debug: false,
5079         dropdownAutoWidth: false,
5080         escapeMarkup: Utils.escapeMarkup,
5081         language: {},
5082         matcher: matcher,
5083         minimumInputLength: 0,
5084         maximumInputLength: 0,
5085         maximumSelectionLength: 0,
5086         minimumResultsForSearch: 0,
5087         selectOnClose: false,
5088         scrollAfterSelect: false,
5089         sorter: function (data) {
5090             return data;
5091         },
5092         templateResult: function (result) {
5093             return result.text;
5094         },
5095         templateSelection: function (selection) {
5096             return selection.text;
5097         },
5098         theme: 'default',
5099         width: 'resolve'
5100     };
5101 };
5102
5103 Defaults.prototype.applyFromElement = function (options, $element) {
5104     var optionLanguage = options.language;
5105     var defaultLanguage = this.defaults.language;
5106     var elementLanguage = $element.prop('lang');
5107     var parentLanguage = $element.closest('[lang]').prop('lang');
5108
5109     var languages = Array.prototype.concat.call(
5110         this._resolveLanguage(elementLanguage),
5111         this._resolveLanguage(optionLanguage),
5112         this._resolveLanguage(defaultLanguage),
5113         this._resolveLanguage(parentLanguage)
5114     );
5115
5116     options.language = languages;
5117
5118     return options;
5119 };
5120
5121 Defaults.prototype._resolveLanguage = function (language) {
5122     if (!language) {
5123         return [];
5124     }
5125
5126     if ($.isEmptyObject(language)) {
5127         return [];
5128     }
5129
5130     if (!$.isPlainObject(language)) {

```

```

5131     return [language];
5132 }
5133
5134 var languages;
5135
5136 if (!$.isArray(language)) {
5137     languages = [language];
5138 } else {
5139     languages = language;
5140 }
5141
5142 var resolvedLanguages = [];
5143
5144 for (var l = 0; l < languages.length; l++) {
5145     resolvedLanguages.push(languages[l]);
5146
5147     if (typeof languages[l] === 'string' && languages[l].indexOf('.') > 0) {
5148         // Extract the region information if it is included
5149         var languageParts = languages[l].split('.');
5150         var baseLanguage = languageParts[0];
5151
5152         resolvedLanguages.push(baseLanguage);
5153     }
5154 }
5155
5156 return resolvedLanguages;
5157 };
5158
5159 Defaults.prototype._processTranslations = function (languages, debug) {
5160     var translations = new Translation();
5161
5162     for (var l = 0; l < languages.length; l++) {
5163         var languageData = new Translation();
5164
5165         var language = languages[l];
5166
5167         if (typeof language === 'string') {
5168             try {
5169                 // Try to load it with the original name
5170                 languageData = Translation.loadPath(language);
5171             } catch (e) {
5172                 try {
5173                     // If we couldn't load it, check if it wasn't the full path
5174                     language = this.defaults.amdLanguageBase + language;
5175                     languageData = Translation.loadPath(language);
5176                 } catch (ex) {
5177                     // The translation could not be loaded at all. Sometimes this is
5178                     // because of a configuration problem, other times this can be
5179                     // because of how Select2 helps load all possible translation files
5180                     if (debug && window.console && console.warn) {
5181                         console.warn(
5182                             'Select2: The language file for "' + language + '" could not ' +
5183                             'not be automatically loaded. A fallback will be used instead.'
5184                         );
5185                     }
5186                 }
5187             }
5188         } else if ($.isPlainObject(language)) {
5189             languageData = new Translation(language);
5190         } else {
5191             languageData = language;
5192         }
5193
5194         translations.extend(languageData);
5195     }
5196
5197     return translations;
5198 };
5199
5200 Defaults.prototype.set = function (key, value) {
5201     var camelKey = $.camelCase(key);
5202
5203     var data = {};
5204     data[camelKey] = value;
5205
5206     var convertedData = Utils._convertData(data);
5207
5208     $.extend(true, this.defaults, convertedData);
5209 };
5210
5211 var defaults = new Defaults();
5212
5213 return defaults;
5214 });
5215
5216 S2.define('select2/options',[
5217     'require',
5218     'jquery',
5219     './defaults',
5220     './utils'
5221 ], function (require, $, Defaults, Utils) {
5222     function Options (options, $element) {
5223         this.options = options;
5224
5225         if ($element != null) {
5226             this.fromElement($element);
5227         }
5228
5229         if ($element != null) {
5230             this.options = Defaults.applyFromElement(this.options, $element);
5231         }
5232
5233         this.options = Defaults.apply(this.options);
5234
5235         if ($element && $element.is('input')) {
5236             var InputCompat = require(this.get('amdBase') + 'compat/inputData');
5237
5238             this.options.dataAdapter = Utils.Decorate(
5239                 this.options.dataAdapter,
5240                 InputCompat
5241             );
5242         }
5243     }
5244
5245     Options.prototype.fromElement = function ($el) {

```



```

5246 var excludedData = ['select2'];
5247
5248 if (this.options.multiple == null) {
5249   this.options.multiple = $.prop('multiple');
5250 }
5251
5252 if (this.options.disabled == null) {
5253   this.options.disabled = $.prop('disabled');
5254 }
5255
5256 if (this.options.dir == null) {
5257   if ($.prop('dir')) {
5258     this.options.dir = $.prop('dir');
5259   } else if ($.closest('[dir]', prop('dir')) {
5260     this.options.dir = $.closest('[dir]').prop('dir');
5261   } else {
5262     this.options.dir = 'ltr';
5263   }
5264 }
5265
5266 $.prop('disabled', this.options.disabled);
5267 $.prop('multiple', this.options.multiple);
5268
5269 if (Utils.GetData($e[0], 'select2Tags')) {
5270   if (this.options.debug && window.console && console.warn) {
5271     console.warn(
5272       'Select2: The `data-select2-tags` attribute has been changed to ' +
5273       'use the `data-data` and `data-tags="true"` attributes and will be ' +
5274       'removed in future versions of Select2.'
5275     );
5276   }
5277
5278   Utils.StoreData($e[0], 'data', Utils.GetData($e[0], 'select2Tags'));
5279   Utils.StoreData($e[0], 'tags', true);
5280 }
5281
5282 if (Utils.GetData($e[0], 'ajaxUrl')) {
5283   if (this.options.debug && window.console && console.warn) {
5284     console.warn(
5285       'Select2: The `data-ajax-url` attribute has been changed to ' +
5286       '`data-ajax-url` and support for the old attribute will be removed' +
5287       ' in future versions of Select2.'
5288     );
5289   }
5290
5291   $.attr('ajax-url', Utils.GetData($e[0], 'ajaxUrl'));
5292   Utils.StoreData($e[0], 'ajax-url', Utils.GetData($e[0], 'ajaxUrl'));
5293 }
5294
5295 var dataset = {};
5296
5297 function upperCaseLetter(_ letter) {
5298   return letter.toUpperCase();
5299 }
5300
5301 // Pre-load all of the attributes which are prefixed with `data-`
5302 for (var attr = 0; attr < $e[0].attributes.length; attr++) {
5303   var attributeName = $e[0].attributes[attr].name;
5304   var prefix = 'data-';
5305
5306   if (attributeName.substr(0, prefix.length) == prefix) {
5307     // Get the contents of the attribute after `data-`
5308     var dataName = attributeName.substr(prefix.length);
5309
5310     // Get the data contents from the consistent source
5311     // This is more than likely the jQuery data helper
5312     var dataValue = Utils.GetData($e[0], dataName);
5313
5314     // camelCase the attribute name to match the spec
5315     var camelDataName = dataName.replace(/-([a-z])/g, upperCaseLetter);
5316
5317     // Store the data attribute contents into the dataset since
5318     dataset[camelDataName] = dataValue;
5319   }
5320 }
5321
5322 // Prefer the element's `dataset` attribute if it exists
5323 // jQuery 1.x does not correctly handle data attributes with multiple dashes
5324 if ($fn.jquery && $fn.jquery.substr(0, 2) == '1.' && $e[0].dataset) {
5325   dataset = $.extend(true, {}, $e[0].dataset, dataset);
5326 }
5327
5328 // Prefer our internal data cache if it exists
5329 var data = $.extend(true, {}, Utils.GetData($e[0]), dataset);
5330
5331 data = Utils._convertData(data);
5332
5333 for (var key in data) {
5334   if ($.inArray(key, excludedData) > -1) {
5335     continue;
5336   }
5337
5338   if ($.isPlainObject(this.options[key])) {
5339     $.extend(this.options[key], data[key]);
5340   } else {
5341     this.options[key] = data[key];
5342   }
5343 }
5344
5345 return this;
5346 };
5347
5348 Options.prototype.get = function (key) {
5349   return this.options[key];
5350 };
5351
5352 Options.prototype.set = function (key, val) {
5353   this.options[key] = val;
5354 };
5355
5356 return Options;
5357 });
5358
5359 S2.define('select2/core', [
5360   'jquery',

```

```

5361 './options',
5362 './utils',
5363 './keys'
5364 ], function ($, Options, Utils, KEYS) {
5365   var Select2 = function (element, options) {
5366     if (Utils.GetData (element[0], 'select2') != null) {
5367       Utils.GetData (element[0], 'select2').destroy();
5368     }
5369
5370     this.$element = $element;
5371
5372     this.id = this._generateId (element);
5373
5374     options = options || {};
5375
5376     this.options = new Options (options, $element);
5377
5378     Select2.__super__.constructor.call (this);
5379
5380     // Set up the tabindex
5381
5382     var tabindex = $element.attr('tabindex') || 0;
5383     Utils.StoreData (element[0], 'old-tabindex', tabindex);
5384     $element.attr('tabindex', '-1');
5385
5386     // Set up containers and adapters
5387
5388     var DataAdapter = this.options.get('dataAdapter');
5389     this.dataAdapter = new DataAdapter (element, this.options);
5390
5391     var $container = this.render();
5392
5393     this._placeContainer ($container);
5394
5395     var SelectionAdapter = this.options.get('selectionAdapter');
5396     this.selection = new SelectionAdapter (element, this.options);
5397     this.$selection = this.selection.render();
5398
5399     this.selection.position (this.$selection, $container);
5400
5401     var DropdownAdapter = this.options.get('dropdownAdapter');
5402     this.dropdown = new DropdownAdapter (element, this.options);
5403     this.$dropdown = this.dropdown.render();
5404
5405     this.dropdown.position (this.$dropdown, $container);
5406
5407     var ResultsAdapter = this.options.get('resultsAdapter');
5408     this.results = new ResultsAdapter (element, this.options, this.dataAdapter);
5409     this.$results = this.results.render();
5410
5411     this.results.position (this.$results, this.$dropdown);
5412
5413     // Bind events
5414
5415     var self = this;
5416
5417     // Bind the container to all of the adapters
5418     this._bindAdapters();
5419
5420     // Register any DOM event handlers
5421     this._registerDomEvents();
5422
5423     // Register any internal event handlers
5424     this._registerDataEvents();
5425     this._registerSelectionEvents();
5426     this._registerDropdownEvents();
5427     this._registerResultsEvents();
5428     this._registerEvents();
5429
5430     // Set the initial state
5431     this.dataAdapter.current (function (initialData) {
5432       self.trigger ('selection:update', {
5433         data: initialData
5434       });
5435     });
5436
5437     // Hide the original select
5438     $element.addClass ('select2-hidden-accessible');
5439     $element.attr ('aria-hidden', 'true');
5440
5441     // Synchronize any monitored attributes
5442     this._syncAttributes();
5443
5444     Utils.StoreData (element[0], 'select2', this);
5445
5446     // Ensure backwards compatibility with $element.data('select2').
5447     $element.data ('select2', this);
5448   },
5449
5450   Utils.Extend (Select2, Utils.Observable);
5451
5452   Select2.prototype._generateId = function (element) {
5453     var id = "";
5454
5455     if ($element.attr('id') != null) {
5456       id = $element.attr('id');
5457     } else if ($element.attr('name') != null) {
5458       id = $element.attr('name') + '-' + Utils.generateChars(2);
5459     } else {
5460       id = Utils.generateChars(4);
5461     }
5462
5463     id = id.replace (/[^\w\d.]*/g, "");
5464     id = 'select2-' + id;
5465
5466     return id;
5467   };
5468
5469   Select2.prototype._placeContainer = function ($container) {
5470     $container.insertAfter (this.$element);
5471
5472     var width = this._resolveWidth (this.$element, this.options.get('width'));
5473
5474     if (width != null) {
5475       $container.css ('width', width);
5476     }
5477
5478     // Determine the width of the container based on the width
5479     // of the element to be replaced
5480     //
5481     // The element width can be determined by trying to get the
5482     // computed width on the element. If that fails, we'll just
5483     // use the element's stated width and assume it's not
5484     // affected by other factors
5485     var styles = window.getComputedStyle ($element[0], null);
5486     var nonComputedWidth = (styles.width != "auto") ? styles.width : null;
5487
5488     if (nonComputedWidth != null) {
5489       width = nonComputedWidth;
5490     } else {
5491       width = $element[0].width;
5492     }
5493
5494     $container.css ('width', width);
5495
5496     // If the container has a class, it's likely already styled
5497     // by the end user
5498     if ($container.hasClass ('select2-container')) {
5499       // If the user has classes on the original ,
5500       // we'll add the same to the new container
5501       $container.addClass ($element.attr ('class'));
5502     }
5503
5504     // Remove the original element
5505     $element.remove();
5506
5507     // Fire the init event
5508     this.trigger ('init');
5509
5510     // Cache the new element
5511     this.$container = $container;
5512
5513     // Fire the ready event
5514     this.trigger ('ready');
5515
5516     // Fire the opened event
5517     this.trigger ('opened');
5518
5519     // Fire the closed event
5520     this.trigger ('closed');
5521
5522     // Fire the destroyed event
5523     this.trigger ('destroyed');
5524
5525     // Fire the destroyed event
5526     this.trigger ('destroyed');
5527
5528     // Fire the destroyed event
5529     this.trigger ('destroyed');
5530
5531     // Fire the destroyed event
5532     this.trigger ('destroyed');
5533
5534     // Fire the destroyed event
5535     this.trigger ('destroyed');
5536
5537     // Fire the destroyed event
5538     this.trigger ('destroyed');
5539
5540     // Fire the destroyed event
5541     this.trigger ('destroyed');
5542
5543     // Fire the destroyed event
5544     this.trigger ('destroyed');
5545
5546     // Fire the destroyed event
5547     this.trigger ('destroyed');
5548
5549     // Fire the destroyed event
5550     this.trigger ('destroyed');
5551
5552     // Fire the destroyed event
5553     this.trigger ('destroyed');
5554
5555     // Fire the destroyed event
5556     this.trigger ('destroyed');
5557
5558     // Fire the destroyed event
5559     this.trigger ('destroyed');
5560
5561     // Fire the destroyed event
5562     this.trigger ('destroyed');
5563
5564     // Fire the destroyed event
5565     this.trigger ('destroyed');
5566
5567     // Fire the destroyed event
5568     this.trigger ('destroyed');
5569
5570     // Fire the destroyed event
5571     this.trigger ('destroyed');
5572
5573     // Fire the destroyed event
5574     this.trigger ('destroyed');
5575
5576     // Fire the destroyed event
5577     this.trigger ('destroyed');
5578
5579     // Fire the destroyed event
5580     this.trigger ('destroyed');
5581
5582     // Fire the destroyed event
5583     this.trigger ('destroyed');
5584
5585     // Fire the destroyed event
5586     this.trigger ('destroyed');
5587
5588     // Fire the destroyed event
5589     this.trigger ('destroyed');
5590
5591     // Fire the destroyed event
5592     this.trigger ('destroyed');
5593
5594     // Fire the destroyed event
5595     this.trigger ('destroyed');
5596
5597     // Fire the destroyed event
5598     this.trigger ('destroyed');
5599
5600     // Fire the destroyed event
5601     this.trigger ('destroyed');
5602
5603     // Fire the destroyed event
5604     this.trigger ('destroyed');
5605
5606     // Fire the destroyed event
5607     this.trigger ('destroyed');
5608
5609     // Fire the destroyed event
5610     this.trigger ('destroyed');
5611
5612     // Fire the destroyed event
5613     this.trigger ('destroyed');
5614
5615     // Fire the destroyed event
5616     this.trigger ('destroyed');
5617
5618     // Fire the destroyed event
5619     this.trigger ('destroyed');
5620
5621     // Fire the destroyed event
5622     this.trigger ('destroyed');
5623
5624     // Fire the destroyed event
5625     this.trigger ('destroyed');
5626
5627     // Fire the destroyed event
5628     this.trigger ('destroyed');
5629
5630     // Fire the destroyed event
5631     this.trigger ('destroyed');
5632
5633     // Fire the destroyed event
5634     this.trigger ('destroyed');
5635
5636     // Fire the destroyed event
5637     this.trigger ('destroyed');
5638
5639     // Fire the destroyed event
5640     this.trigger ('destroyed');
5641
5642     // Fire the destroyed event
5643     this.trigger ('destroyed');
5644
5645     // Fire the destroyed event
5646     this.trigger ('destroyed');
5647
5648     // Fire the destroyed event
5649     this.trigger ('destroyed');
5650
5651     // Fire the destroyed event
5652     this.trigger ('destroyed');
5653
5654     // Fire the destroyed event
5655     this.trigger ('destroyed');
5656
5657     // Fire the destroyed event
5658     this.trigger ('destroyed');
5659
5660     // Fire the destroyed event
5661     this.trigger ('destroyed');
5662
5663     // Fire the destroyed event
5664     this.trigger ('destroyed');
5665
5666     // Fire the destroyed event
5667     this.trigger ('destroyed');
5668
5669     // Fire the destroyed event
5670     this.trigger ('destroyed');
5671
5672     // Fire the destroyed event
5673     this.trigger ('destroyed');
5674
5675     // Fire the destroyed event
5676     this.trigger ('destroyed');
5677
5678     // Fire the destroyed event
5679     this.trigger ('destroyed');
5680
5681     // Fire the destroyed event
5682     this.trigger ('destroyed');
5683
5684     // Fire the destroyed event
5685     this.trigger ('destroyed');
5686
5687     // Fire the destroyed event
5688     this.trigger ('destroyed');
5689
5690     // Fire the destroyed event
5691     this.trigger ('destroyed');
5692
5693     // Fire the destroyed event
5694     this.trigger ('destroyed');
5695
5696     // Fire the destroyed event
5697     this.trigger ('destroyed');
5698
5699     // Fire the destroyed event
5700     this.trigger ('destroyed');
5701
5702     // Fire the destroyed event
5703     this.trigger ('destroyed');
5704
5705     // Fire the destroyed event
5706     this.trigger ('destroyed');
5707
5708     // Fire the destroyed event
5709     this.trigger ('destroyed');
5710
5711     // Fire the destroyed event
5712     this.trigger ('destroyed');
5713
5714     // Fire the destroyed event
5715     this.trigger ('destroyed');
5716
5717     // Fire the destroyed event
5718     this.trigger ('destroyed');
5719
5720     // Fire the destroyed event
5721     this.trigger ('destroyed');
5722
5723     // Fire the destroyed event
5724     this.trigger ('destroyed');
5725
5726     // Fire the destroyed event
5727     this.trigger ('destroyed');
5728
5729     // Fire the destroyed event
5730     this.trigger ('destroyed');
5731
5732     // Fire the destroyed event
5733     this.trigger ('destroyed');
5734
5735     // Fire the destroyed event
5736     this.trigger ('destroyed');
5737
5738     // Fire the destroyed event
5739     this.trigger ('destroyed');
5740
5741     // Fire the destroyed event
5742     this.trigger ('destroyed');
5743
5744     // Fire the destroyed event
5745     this.trigger ('destroyed');
5746
5747     // Fire the destroyed event
5748     this.trigger ('destroyed');
5749
5750     // Fire the destroyed event
5751     this.trigger ('destroyed');
5752
5753     // Fire the destroyed event
5754     this.trigger ('destroyed');
5755
5756     // Fire the destroyed event
5757     this.trigger ('destroyed');
5758
5759     // Fire the destroyed event
5760     this.trigger ('destroyed');
5761
5762     // Fire the destroyed event
5763     this.trigger ('destroyed
```

```

5476 }
5477 };
5478
5479 Select2.prototype._resolveWidth = function ($element, method) {
5480     var WIDTH = /^width:(([-+]?([0-9]*\.)?[0-9]+)(px|em|ex|%|in|cm|mm|pt|pc))/i;
5481
5482     if (method == 'resolve') {
5483         var styleWidth = this._resolveWidth($element, 'style');
5484
5485         if (styleWidth != null) {
5486             return styleWidth;
5487         }
5488
5489         return this._resolveWidth($element, 'element');
5490     }
5491
5492     if (method == 'element') {
5493         var elementWidth = $element.outerWidth(false);
5494
5495         if (elementWidth <= 0) {
5496             return 'auto';
5497         }
5498
5499         return elementWidth + 'px';
5500     }
5501
5502     if (method == 'style') {
5503         var style = $element.attr('style');
5504
5505         if (typeof style != 'string') {
5506             return null;
5507         }
5508
5509         var attrs = style.split(';');
5510
5511         for (var i = 0, l = attrs.length, i < l; i = i + 1) {
5512             var attr = attrs[i].replace(/\s/g, "");
5513             var matches = attr.match(WIDTH);
5514
5515             if (matches != null && matches.length >= 1) {
5516                 return matches[1];
5517             }
5518         }
5519
5520         return null;
5521     }
5522
5523     if (method == 'computedStyle') {
5524         var computedStyle = window.getComputedStyle($element[0]);
5525
5526         return computedStyle.width;
5527     }
5528
5529     return method;
5530 };
5531
5532 Select2.prototype._bindAdapters = function () {
5533     this.dataAdapter.bind(this, this.$container);
5534     this.selection.bind(this, this.$container);
5535
5536     this.dropdown.bind(this, this.$container);
5537     this.results.bind(this, this.$container);
5538 };
5539
5540 Select2.prototype._registerDomEvents = function () {
5541     var self = this;
5542
5543     this.$element.on('change.select2', function () {
5544         self.dataAdapter.current(function (data) {
5545             self.trigger('selection:update', {
5546                 data: data
5547             });
5548         });
5549     });
5550
5551     this.$element.on('focus.select2', function (evt) {
5552         self.trigger('focus', evt);
5553     });
5554
5555     this._syncA = Utils.bind(this._syncAttributes, this);
5556     this._syncS = Utils.bind(this._syncSubtree, this);
5557
5558     if (this.$element[0].attachEvent) {
5559         this.$element[0].attachEvent('onpropertychange', this._syncA);
5560     }
5561
5562     var observer = window.MutationObserver ||
5563         window.WebKitMutationObserver ||
5564         window.MozMutationObserver
5565     ;
5566
5567     if (observer != null) {
5568         this._observer = new observer(function (mutations) {
5569             self._syncA();
5570             self._syncS(null, mutations);
5571         });
5572         this._observer.observe(this.$element[0], {
5573             attributes: true,
5574             childList: true,
5575             subtree: false
5576         });
5577     } else if (this.$element[0].addEventListener) {
5578         this.$element[0].addEventListener(
5579             'DOMAttrModified',
5580             self._syncA,
5581             false
5582         );
5583         this.$element[0].addEventListener(
5584             'DOMNodeInserted',
5585             self._syncS,
5586             false
5587         );
5588         this.$element[0].addEventListener(
5589             'DOMNodeRemoved',
5590             self._syncS
5591         );
5592     }
5593 }

```

```

5591     false
5592   };
5593 }
5594 };
5595
5596 Select2.prototype._registerDataEvents = function () {
5597   var self = this;
5598
5599   this.dataAdapter.on('*', function (name, params) {
5600     self.trigger(name, params);
5601   });
5602 };
5603
5604 Select2.prototype._registerSelectionEvents = function () {
5605   var self = this;
5606   var nonRelayEvents = ['toggle', 'focus'];
5607
5608   this.selection.on('toggle', function () {
5609     self.toggleDropdown();
5610   });
5611
5612   this.selection.on('focus', function (params) {
5613     self.focus(params);
5614   });
5615
5616   this.selection.on('*', function (name, params) {
5617     if ($.inArray(name, nonRelayEvents) !== -1) {
5618       return;
5619     }
5620
5621     self.trigger(name, params);
5622   });
5623 };
5624
5625 Select2.prototype._registerDropdownEvents = function () {
5626   var self = this;
5627
5628   this.dropdown.on('*', function (name, params) {
5629     self.trigger(name, params);
5630   });
5631 };
5632
5633 Select2.prototype._registerResultsEvents = function () {
5634   var self = this;
5635
5636   this.results.on('*', function (name, params) {
5637     self.trigger(name, params);
5638   });
5639 };
5640
5641 Select2.prototype._registerEvents = function () {
5642   var self = this;
5643
5644   this.on('open', function () {
5645     self.$container.addClass('select2-container--open');
5646   });
5647
5648   this.on('close', function () {
5649     self.$container.removeClass('select2-container--open');
5650   });
5651
5652   this.on('enable', function () {
5653     self.$container.removeClass('select2-container--disabled');
5654   });
5655
5656   this.on('disable', function () {
5657     self.$container.addClass('select2-container--disabled');
5658   });
5659
5660   this.on('blur', function () {
5661     self.$container.removeClass('select2-container--focus');
5662   });
5663
5664   this.on('query', function (params) {
5665     if (!self.isOpen()) {
5666       self.trigger('open', {});
5667     }
5668
5669     this.dataAdapter.query(params, function (data) {
5670       self.trigger('results:all', {
5671         data: data,
5672         query: params
5673       });
5674     });
5675   });
5676
5677   this.on('query:append', function (params) {
5678     this.dataAdapter.query(params, function (data) {
5679       self.trigger('results:append', {
5680         data: data,
5681         query: params
5682       });
5683     });
5684   });
5685
5686   this.on('keypress', function (evt) {
5687     var key = evt.which;
5688
5689     if (self.isOpen()) {
5690       if (key === KEYS.ESC || key === KEYS.TAB ||
5691         (key === KEYS.UP && evt.altKey)) {
5692         self.close(evt);
5693
5694         evt.preventDefault();
5695       } else if (key === KEYS.ENTER) {
5696         self.trigger('results:select', {});
5697
5698         evt.preventDefault();
5699       } else if ((key === KEYS.SPACE && evt.ctrlKey)) {
5700         self.trigger('results:toggle', {});
5701
5702         evt.preventDefault();
5703       } else if (key === KEYS.UP) {
5704         self.trigger('results:previous', {});
5705       }
5706     }
5707   });

```

```

5706     evt.preventDefault();
5707 } else if (key === KEYS.DOWN) {
5708     self.trigger('results:next', {});
5709
5710     evt.preventDefault();
5711 }
5712 } else {
5713     if (key === KEYS.ENTER || key === KEYS.SPACE ||
5714         key === KEYS.DOWN && evt.altKey) {
5715         self.open();
5716
5717         evt.preventDefault();
5718     }
5719 }
5720 });
5721 };
5722
5723 Select2.prototype._syncAttributes = function () {
5724     this.options.set('disabled', this.$element.prop('disabled'));
5725
5726     if (this.isDisabled()) {
5727         if (this.isOpen()) {
5728             this.close();
5729         }
5730
5731         this.trigger('disable', {});
5732     } else {
5733         this.trigger('enable', {});
5734     }
5735 };
5736
5737 Select2.prototype._isChangeMutation = function (evt, mutations) {
5738     var changed = false;
5739     var self = this;
5740
5741     // Ignore any mutation events raised for elements that aren't options or
5742     // optgroups. This handles the case when the select element is destroyed
5743     if (
5744         evt && evt.target && (
5745             evt.target.nodeName !== 'OPTION' && evt.target.nodeName !== 'OPTGROUP'
5746         )
5747     ) {
5748         return;
5749     }
5750
5751     if (!mutations) {
5752         // If mutation events aren't supported, then we can only assume that the
5753         // change affected the selections
5754         changed = true;
5755     } else if (mutations.addedNodes && mutations.addedNodes.length > 0) {
5756         for (var n = 0; n < mutations.addedNodes.length; n++) {
5757             var node = mutations.addedNodes[n];
5758
5759             if (node.selected) {
5760                 changed = true;
5761             }
5762         }
5763     } else if (mutations.removedNodes && mutations.removedNodes.length > 0) {
5764         changed = true;
5765     } else if ($.isArray(mutations)) {
5766         $.each(mutations, function (evt, mutation) {
5767             if (self._isChangeMutation(evt, mutation)) {
5768                 // We've found a change mutation.
5769                 // Let's escape from the loop and continue
5770                 changed = true;
5771                 return false;
5772             }
5773         });
5774     }
5775     return changed;
5776 };
5777
5778 Select2.prototype._syncSubtree = function (evt, mutations) {
5779     var changed = this._isChangeMutation(evt, mutations);
5780     var self = this;
5781
5782     // Only re-pull the data if we think there is a change
5783     if (changed) {
5784         this.dataAdapter.current(function (currentData) {
5785             self.trigger('selection:update', {
5786                 data: currentData
5787             });
5788         });
5789     }
5790 };
5791
5792 /**
5793  * Override the trigger method to automatically trigger pre-events when
5794  * there are events that can be prevented.
5795  */
5796 Select2.prototype.trigger = function (name, args) {
5797     var actualTrigger = Select2.__super__.trigger;
5798     var preTriggerMap = {
5799         'open': 'opening',
5800         'close': 'closing',
5801         'select': 'selecting',
5802         'unselect': 'unselecting',
5803         'clear': 'clearing'
5804     };
5805
5806     if (args === undefined) {
5807         args = {};
5808     }
5809
5810     if (name in preTriggerMap) {
5811         var preTriggerName = preTriggerMap[name];
5812         var preTriggerArgs = {
5813             prevented: false,
5814             name: name,
5815             args: args
5816         };
5817
5818         actualTrigger.call(this, preTriggerName, preTriggerArgs);
5819
5820         if (preTriggerArgs.prevented) {

```

```

5821     args prevented = true;
5822
5823     return;
5824 }
5825 }
5826
5827 actualTrigger.call(this, name, args);
5828 };
5829
5830 Select2.prototype.toggleDropdown = function () {
5831     if (this.isDisabled()) {
5832         return;
5833     }
5834
5835     if (this.isOpen()) {
5836         this.close();
5837     } else {
5838         this.open();
5839     }
5840 };
5841
5842 Select2.prototype.open = function () {
5843     if (this.isOpen()) {
5844         return;
5845     }
5846
5847     if (this.isDisabled()) {
5848         return;
5849     }
5850
5851     this.trigger('query', {});
5852 };
5853
5854 Select2.prototype.close = function (evt) {
5855     if (!this.isOpen()) {
5856         return;
5857     }
5858
5859     this.trigger('close', { originalEvent: evt });
5860 };
5861
5862 /**
5863  * Helper method to abstract the "enabled" (not "disabled") state of this
5864  * object.
5865  *
5866  * @return {true} if the instance is not disabled.
5867  * @return {false} if the instance is disabled.
5868  */
5869 Select2.prototype.isEnabled = function () {
5870     return !this.isDisabled();
5871 };
5872
5873 /**
5874  * Helper method to abstract the "disabled" state of this object.
5875  *
5876  * @return {true} if the disabled option is true.
5877  * @return {false} if the disabled option is false.
5878  */
5879 Select2.prototype.isDisabled = function () {
5880     return this.options.get('disabled');
5881 };
5882
5883 Select2.prototype.isOpen = function () {
5884     return this.$container.hasClass('select2-container--open');
5885 };
5886
5887 Select2.prototype.hasFocus = function () {
5888     return this.$container.hasClass('select2-container--focus');
5889 };
5890
5891 Select2.prototype.focus = function (data) {
5892     // No need to re-trigger focus events if we are already focused
5893     if (this.hasFocus()) {
5894         return;
5895     }
5896
5897     this.$container.addClass('select2-container--focus');
5898     this.trigger('focus', {});
5899 };
5900
5901 Select2.prototype.enable = function (args) {
5902     if (this.options.get('debug') && window.console && console.warn) {
5903         console.warn(
5904             'Select2: The `select2("enable")` method has been deprecated and will ' +
5905             'be removed in later Select2 versions. Use `element.prop("disabled")` ' +
5906             'instead.'
5907         );
5908     }
5909
5910     if (args == null || args.length === 0) {
5911         args = [true];
5912     }
5913
5914     var disabled = !args[0];
5915
5916     this.$element.prop('disabled', disabled);
5917 };
5918
5919 Select2.prototype.data = function () {
5920     if (this.options.get('debug') &&
5921         arguments.length > 0 && window.console && console.warn) {
5922         console.warn(
5923             'Select2: Data can no longer be set using `select2("data")`. You ' +
5924             'should consider setting the value instead using `element.val()`. '
5925         );
5926     }
5927
5928     var data = [];
5929
5930     this.dataAdapter.current(function (currentData) {
5931         data = currentData;
5932     });
5933
5934     return data;
5935 };

```

```

5936
5937 Select2.prototype.val = function (args) {
5938   if (this.options.get('debug') && window.console && console.warn) {
5939     console.warn(
5940       'Select2: The `select2("val")` method has been deprecated and will be ' +
5941       'removed in later Select2 versions. Use $element.val() instead.'
5942     );
5943   }
5944
5945   if (args == null || args.length === 0) {
5946     return this.$element.val();
5947   }
5948
5949   var newVal = args[0];
5950
5951   if ($.isArray(newVal)) {
5952     newVal = $.map(newVal, function (obj) {
5953       return obj.toString();
5954     });
5955   }
5956
5957   this.$element.val(newVal).trigger('input').trigger('change');
5958 };
5959
5960 Select2.prototype.destroy = function () {
5961   this.$container.remove();
5962
5963   if (this.$element[0].detachEvent) {
5964     this.$element[0].detachEvent('onpropertychange', this._syncA);
5965   }
5966
5967   if (this._observer != null) {
5968     this._observer.disconnect();
5969     this._observer = null;
5970   } else if (this.$element[0].removeEventListener) {
5971     this.$element[0].removeEventListener('DOMAttrModified', this._syncA, false);
5972     this.$element[0].removeEventListener('DOMNodeInserted', this._syncS, false);
5973     this.$element[0].removeEventListener('DOMNodeRemoved', this._syncS, false);
5974   }
5975
5976   this._syncA = null;
5977   this._syncS = null;
5978
5979   this.$element.off('.select2');
5980   this.$element.attr('tabindex',
5981     Utils.GetData(this.$element[0], 'old-tabindex'));
5982
5983   this.$element.removeClass('select2-hidden-accessible');
5984   this.$element.attr('aria-hidden', 'false');
5985   Utils.RemoveData(this.$element[0]);
5986   this.$element.removeData('select2');
5987
5988   this.dataAdapter.destroy();
5989   this.selection.destroy();
5990   this.dropdown.destroy();
5991   this.results.destroy();
5992
5993   this.dataAdapter = null;
5994   this.selection = null;
5995   this.dropdown = null;
5996   this.results = null;
5997 };
5998
5999 Select2.prototype.render = function () {
6000   var $container = $(
6001     '<span class="select2 select2-container">' +
6002     '<span class="selection"></span>' +
6003     '<span class="dropdown-wrapper" aria-hidden="true"></span>' +
6004     '</span>'
6005   );
6006
6007   $container.attr('dir', this.options.get('dir'));
6008
6009   this.$container = $container;
6010
6011   this.$container.addClass('select2-container-' + this.options.get('theme'));
6012
6013   Utils.StoreData($container[0], 'element', this.$element);
6014
6015   return $container;
6016 };
6017
6018 return Select2;
6019 };
6020
6021 // Select2
6022 S2.define('select2/compat/utils',[
6023   'jquery'
6024 ], function ($) {
6025   function syncCssClasses ($dest, $src, adapter) {
6026     var classes_replacements = [], adapted
6027
6028     classes = $.trim($dest.attr('class'));
6029
6030     if (classes) {
6031       classes = "" + classes; // for IE which returns object
6032
6033       $.classes.split(/\s+/).each(function () {
6034         // Save all Select2 classes
6035         if (this.indexOf('select2-') === 0) {
6036           replacements.push(this);
6037         }
6038       });
6039     }
6040
6041     classes = $.trim($src.attr('class'));
6042
6043     if (classes) {
6044       classes = "" + classes; // for IE which returns object
6045
6046       $.classes.split(/\s+/).each(function () {
6047         // Only adapt non-Select2 classes
6048         if (this.indexOf('select2-') !== 0) {
6049

```

```

6051     adapted = adapter(this);
6052
6053     if (adapted != null) {
6054         replacements.push(adapted);
6055     }
6056 }
6057 });
6058 }
6059
6060 $dest.attr('class', replacements.join(' '));
6061 }
6062
6063 return {
6064     syncCssClasses: syncCssClasses
6065 };
6066 });
6067
6068 S2.define('select2/compat/containerCss',[
6069     'jquery',
6070     './utils'
6071 ],function($, CompatUtils) {
6072     // No-op CSS adapter that discards all classes by default
6073     function _containerAdapter (clazz) {
6074         return null;
6075     }
6076
6077     function ContainerCSS () { }
6078
6079     ContainerCSS.prototype.render = function (decorated) {
6080         var $container = decorated.call(this);
6081
6082         var containerCssClass = this.options.get('containerCssClass') || '';
6083
6084         if ($.isFunction(containerCssClass)) {
6085             containerCssClass = containerCssClass(this.$element);
6086         }
6087
6088         var containerCssAdapter = this.options.get('adaptContainerCssClass');
6089         containerCssAdapter = containerCssAdapter || _containerAdapter;
6090
6091         if (containerCssClass.indexOf(':all:') !== -1) {
6092             containerCssClass = containerCssClass.replace(':all:', '');
6093
6094             var _cssAdapter = containerCssAdapter;
6095
6096             containerCssAdapter = function (clazz) {
6097                 var adapted = _cssAdapter(clazz);
6098
6099                 if (adapted != null) {
6100                     // Append the old one along with the adapted one
6101                     return adapted + ' ' + clazz;
6102                 }
6103
6104                 return clazz;
6105             };
6106         }
6107
6108         var containerCss = this.options.get('containerCss') || {};
6109
6110         if ($.isFunction(containerCss)) {
6111             containerCss = containerCss(this.$element);
6112         }
6113
6114         CompatUtils.syncCssClasses($container, this.$element, containerCssAdapter);
6115
6116         $container.css(containerCss);
6117         $container.addClass(containerCssClass);
6118
6119         return $container;
6120     };
6121
6122     return ContainerCSS;
6123 });
6124
6125 S2.define('select2/compat/dropdownCss',[
6126     'jquery',
6127     './utils'
6128 ],function($, CompatUtils) {
6129     // No-op CSS adapter that discards all classes by default
6130     function _dropdownAdapter (clazz) {
6131         return null;
6132     }
6133
6134     function DropdownCSS () { }
6135
6136     DropdownCSS.prototype.render = function (decorated) {
6137         var $dropdown = decorated.call(this);
6138
6139         var dropdownCssClass = this.options.get('dropdownCssClass') || '';
6140
6141         if ($.isFunction(dropdownCssClass)) {
6142             dropdownCssClass = dropdownCssClass(this.$element);
6143         }
6144
6145         var dropdownCssAdapter = this.options.get('adaptDropdownCssClass');
6146         dropdownCssAdapter = dropdownCssAdapter || _dropdownAdapter;
6147
6148         if (dropdownCssClass.indexOf(':all:') !== -1) {
6149             dropdownCssClass = dropdownCssClass.replace(':all:', '');
6150
6151             var _cssAdapter = dropdownCssAdapter;
6152
6153             dropdownCssAdapter = function (clazz) {
6154                 var adapted = _cssAdapter(clazz);
6155
6156                 if (adapted != null) {
6157                     // Append the old one along with the adapted one
6158                     return adapted + ' ' + clazz;
6159                 }
6160
6161                 return clazz;
6162             };
6163         }
6164
6165         var dropdownCss = this.options.get('dropdownCss') || {};

```



```

6166
6167 if ($ isFunction dropdownCss) {
6168     dropdownCss = dropdownCss(this.$element);
6169 }
6170
6171 CompatUtils syncCssClasses($dropdown, this.$element, dropdownCssAdapter);
6172
6173 $dropdown.css(dropdownCss);
6174 $dropdown.addClass(dropdownCssClass);
6175
6176 return $dropdown;
6177 };
6178
6179 return DropdownCSS;
6180 });
6181
6182 S2.define('select2/compat/initSelection',[
6183     'jquery'
6184 ],function ($) {
6185     function InitSelection(decorated,$element,options){
6186         if(options.get('debug') && window.console && console.warn){
6187             console.warn(
6188                 'Select2: The `initSelection` option has been deprecated in favor of ' +
6189                 'a custom data adapter that overrides the `current` method.' +
6190                 'This method is now called multiple times instead of a single ' +
6191                 'time when the instance is initialized. Support will be removed ' +
6192                 'for the `initSelection` option in future versions of Select2'
6193             );
6194         }
6195
6196         this.initSelection = options.get('initSelection');
6197         this._initialized = false;
6198
6199         decorated.call(this,$element,options);
6200     }
6201
6202     InitSelection.prototype.current = function(decorated,callback){
6203         var self = this;
6204
6205         if(this._initialized){
6206             decorated.call(this,callback);
6207
6208             return;
6209         }
6210
6211         this.initSelection.call(null,this.$element,function(data){
6212             self._initialized = true;
6213
6214             if(!$.isArray(data)){
6215                 data = [data];
6216             }
6217
6218             callback(data);
6219         });
6220     };
6221
6222     return InitSelection;
6223 });
6224
6225 S2.define('select2/compat/inputData',[
6226     'jquery',
6227     './utils'
6228 ],function ($,Utils){
6229     function InputData(decorated,$element,options){
6230         this._currentData = [];
6231         this._valueSeparator = options.get('valueSeparator') || ',';
6232
6233         if($element.prop('type') === 'hidden'){
6234             if(options.get('debug') && console && console.warn){
6235                 console.warn(
6236                     'Select2: Using a hidden input with Select2 is no longer ' +
6237                     'supported and may stop working in the future. It is recommended ' +
6238                     'to use a `<select>` element instead.'
6239                 );
6240             }
6241         }
6242
6243         decorated.call(this,$element,options);
6244     }
6245
6246     InputData.prototype.current = function(_ ,callback){
6247         function getSelected(data,selectedIds){
6248             var selected = [];
6249
6250             if(data.selected || $.inArray(data.id,selectedIds) !== -1){
6251                 data.selected = true;
6252                 selected.push(data);
6253             } else {
6254                 data.selected = false;
6255             }
6256
6257             if(data.children){
6258                 selected.push.apply(selected,getSelected(data.children,selectedIds));
6259             }
6260
6261             return selected;
6262         }
6263
6264         var selected = [];
6265
6266         for(var d = 0; d < this._currentData.length; d++){
6267             var data = this._currentData[d];
6268
6269             selected.push.apply(
6270                 selected,
6271                 getSelected(
6272                     data,
6273                     this.$element.val().split(
6274                         this._valueSeparator
6275                     )
6276                 )
6277             );
6278         }
6279
6280         callback(selected);

```

```

6281 };
6282
6283 InputData.prototype.select = function (_ data) {
6284   if (!this.options.get('multiple')) {
6285     this.current(function (allData) {
6286       $map(allData, function (data) {
6287         data.selected = false;
6288       });
6289     });
6290
6291     this.$element.val(data.id);
6292     this.$element.trigger('input').trigger('change');
6293   } else {
6294     var value = this.$element.val();
6295     value += this._valueSeparator + data.id;
6296
6297     this.$element.val(value);
6298     this.$element.trigger('input').trigger('change');
6299   }
6300 };
6301
6302 InputData.prototype.unselect = function (_ data) {
6303   var self = this;
6304
6305   data.selected = false;
6306
6307   this.current(function (allData) {
6308     var values = [];
6309
6310     for (var d = 0; d < allData.length; d++) {
6311       var item = allData[d];
6312
6313       if (data.id == item.id) {
6314         continue;
6315       }
6316
6317       values.push(item.id);
6318     }
6319
6320     self.$element.val(values.join(self._valueSeparator));
6321     self.$element.trigger('input').trigger('change');
6322   });
6323 };
6324
6325 InputData.prototype.query = function (_ params, callback) {
6326   var results = [];
6327
6328   for (var d = 0; d < this._currentData.length; d++) {
6329     var data = this._currentData[d];
6330
6331     var matches = this.matches(params, data);
6332
6333     if (matches !== null) {
6334       results.push(matches);
6335     }
6336   }
6337
6338   callback({
6339     results: results
6340   });
6341 };
6342
6343 InputData.prototype.addOptions = function (_ $options) {
6344   var options = $.map($options, function ($option) {
6345     return Utils.GetData($option[0], 'data');
6346   });
6347
6348   this._currentData.push.apply(this._currentData, options);
6349 };
6350
6351 return InputData;
6352 });
6353
6354 S2.define('select2/compat/matcher', [
6355   'jquery'
6356 ], function ($) {
6357   function oldMatcher (matcher) {
6358     function wrappedMatcher (params, data) {
6359       var match = $.extend(true, {}, data);
6360
6361       if (params.term == null || $.trim(params.term) === '') {
6362         return match;
6363       }
6364
6365       if (data.children) {
6366         for (var c = data.children.length - 1; c >= 0; c--) {
6367           var child = data.children[c];
6368
6369           // Check if the child object matches
6370           // The old matcher returned a boolean true or false
6371           var doesMatch = matcher(params.term, child.text, child);
6372
6373           // If the child didn't match, pop it off
6374           if (!doesMatch) {
6375             match.children.splice(c, 1);
6376           }
6377         }
6378
6379         if (match.children.length > 0) {
6380           return match;
6381         }
6382       }
6383
6384       if (matcher(params.term, data.text, data)) {
6385         return match;
6386       }
6387
6388       return null;
6389     }
6390
6391     return wrappedMatcher;
6392   }
6393
6394   return oldMatcher;
6395 });

```

```

6396
6397 S2 define('select2/compat/query',[
6398
6399 ],function () {
6400   function Query (decorated, $element, options) {
6401     if (options.get('debug') && window.console && console.warn) {
6402       console.warn(
6403         'Select2: The `query` option has been deprecated in favor of a `+
6404         `custom data adapter that overrides the `query` method. Support `+
6405         `will be removed for the `query` option in future versions of `+
6406         `Select2.'
6407       );
6408     }
6409
6410     decorated.call(this, $element, options);
6411   }
6412
6413   Query.prototype.query = function (_, params, callback) {
6414     params.callback = callback;
6415
6416     var query = this.options.get('query');
6417
6418     query.call(null, params);
6419   };
6420
6421   return Query;
6422 });
6423
6424 S2 define('select2/dropdown/attachContainer',[
6425
6426 ],function () {
6427   function AttachContainer (decorated, $element, options) {
6428     decorated.call(this, $element, options);
6429   }
6430
6431   AttachContainer.prototype.position =
6432   function (decorated, $dropdown, $container) {
6433     var $dropdownContainer = $container.find('.dropdown-wrapper');
6434     $dropdownContainer.append($dropdown);
6435
6436     $dropdown.addClass('select2-dropdown--below');
6437     $container.addClass('select2-container--below');
6438   };
6439
6440   return AttachContainer;
6441 });
6442
6443 S2 define('select2/dropdown/stopPropagation',[
6444
6445 ],function () {
6446   function StopPropagation () { }
6447
6448   StopPropagation.prototype.bind = function (decorated, container, $container) {
6449     decorated.call(this, container, $container);
6450
6451     var stoppedEvents = [
6452       'blur',
6453       'change',
6454       'click',
6455       'dblclick',
6456       'focus',
6457       'focusin',
6458       'focusout',
6459       'input',
6460       'keydown',
6461       'keyup',
6462       'keypress',
6463       'mousedown',
6464       'mouseenter',
6465       'mouseleave',
6466       'mousemove',
6467       'mouseover',
6468       'mouseup',
6469       'search',
6470       'touchend',
6471       'touchstart'
6472     ];
6473
6474     this.$dropdown.on(stoppedEvents.join(' '), function (evt) {
6475       evt.stopPropagation();
6476     });
6477   };
6478
6479   return StopPropagation;
6480 });
6481
6482 S2 define('select2/selection/stopPropagation',[
6483
6484 ],function () {
6485   function StopPropagation () { }
6486
6487   StopPropagation.prototype.bind = function (decorated, container, $container) {
6488     decorated.call(this, container, $container);
6489
6490     var stoppedEvents = [
6491       'blur',
6492       'change',
6493       'click',
6494       'dblclick',
6495       'focus',
6496       'focusin',
6497       'focusout',
6498       'input',
6499       'keydown',
6500       'keyup',
6501       'keypress',
6502       'mousedown',
6503       'mouseenter',
6504       'mouseleave',
6505       'mousemove',
6506       'mouseover',
6507       'mouseup',
6508       'search',
6509       'touchend',
6510       'touchstart'

```

```

6511 };
6512
6513 this.$selection.on('stoppedEvents join!', function (evt) {
6514     evt.stopPropagation();
6515 });
6516 };
6517
6518 return StopPropagation;
6519 });
6520
6521 /*!
6522 * jQuery Mousewheel 3.1.13
6523 *
6524 * Copyright jQuery Foundation and other contributors
6525 * Released under the MIT license
6526 * http://jquery.org/license
6527 */
6528
6529 (function (factory) {
6530     if (typeof S2.define === 'function' && S2.define.amd) {
6531         // AMD, Register as an anonymous module.
6532         S2.define('jquery-mousewheel', ['jquery'], factory);
6533     } else if (typeof exports === 'object') {
6534         // Node/CommonJS style for Browserify
6535         module.exports = factory;
6536     } else {
6537         // Browser globals
6538         factory(jQuery);
6539     }
6540 })(function ($) {
6541
6542     var toFix = ['wheel', 'mousewheel', 'DOMMouseScroll', 'MozMousePixelScroll'],
6543         toBind = ('onwheel' in document || document.documentMode >= 9) ?
6544             ['wheel'] : ['mousewheel', 'DomMouseScroll', 'MozMousePixelScroll'],
6545         slice = Array.prototype.slice,
6546         nullLowestDeltaTimeout, lowestDelta;
6547
6548     if ( $.event.fixHooks ) {
6549         for ( var i = toFix.length; i; ) {
6550             $.event.fixHooks[ toFix[--i] ] = $.event.mouseHooks;
6551         }
6552     }
6553
6554     var special = $.event.special.mousewheel = {
6555         version: '3.1.12',
6556
6557         setup: function() {
6558             if ( this.addEventListener ) {
6559                 for ( var i = toBind.length; i; ) {
6560                     this.addEventListener( toBind[--i], handler, false );
6561                 }
6562             } else {
6563                 this.onmousewheel = handler;
6564             }
6565             // Store the line height and page height for this particular element
6566             $.data(this, 'mousewheel-line-height', special.getLineHeight(this));
6567             $.data(this, 'mousewheel-page-height', special.getPageHeight(this));
6568         },
6569
6570         teardown: function() {
6571             if ( this.removeEventListener ) {
6572                 for ( var i = toBind.length; i; ) {
6573                     this.removeEventListener( toBind[--i], handler, false );
6574                 }
6575             } else {
6576                 this.onmousewheel = null;
6577             }
6578             // Clean up the data we added to the element
6579             $.removeData(this, 'mousewheel-line-height');
6580             $.removeData(this, 'mousewheel-page-height');
6581         },
6582
6583         getLineHeight: function(elem) {
6584             var $elem = $(elem),
6585                 $parent = $elem.offsetParent() in $fn ? 'offsetParent' : 'parent'();
6586             if ( $parent.length ) {
6587                 $parent = $('body');
6588             }
6589             return parseInt($parent.css('fontSize'), 10) || parseInt($elem.css('fontSize'), 10) || 16;
6590         },
6591
6592         getPageHeight: function(elem) {
6593             return $(elem).height();
6594         },
6595
6596         settings: {
6597             adjustOldDeltas: true, // see shouldAdjustOldDeltas() below
6598             normalizeOffset: true // calls getBoundingClientRect for each event
6599         }
6600     };
6601
6602     $.fn.extend({
6603         mousewheel: function(fn) {
6604             return fn ? this.bind('mousewheel', fn) : this.trigger('mousewheel');
6605         },
6606
6607         unmousewheel: function(fn) {
6608             return this.unbind('mousewheel', fn);
6609         }
6610     });
6611
6612     function handler(event) {
6613         var orgEvent = event || window.event,
6614             args = slice.call(arguments, 1),
6615             delta = 0,
6616             deltaX = 0,
6617             deltaY = 0,
6618             absDelta = 0,
6619             offsetX = 0,
6620             offsetY = 0;
6621         event = $.event.fix(orgEvent);
6622         event.type = 'mousewheel';
6623
6624         // Old school scrollwheel delta
6625     
```

```

6626 if ( 'detail' in orgEvent ) { deltaY = orgEvent.detail * -1; }
6627 if ( 'wheelDelta' in orgEvent ) { deltaY = orgEvent.wheelDelta; }
6628 if ( 'wheelDeltaY' in orgEvent ) { deltaY = orgEvent.wheelDeltaY; }
6629 if ( 'wheelDeltaX' in orgEvent ) { deltaX = orgEvent.wheelDeltaX * -1; }
6630
6631 // Firefox < 17 horizontal scrolling related to DOMMouseScroll event
6632 if ( 'axis' in orgEvent && orgEvent.axis === orgEvent.HORIZONTAL_AXIS ) {
6633     deltaX = deltaY * -1;
6634     deltaY = 0;
6635 }
6636
6637 // Set delta to be deltaY or deltaX if deltaY is 0 for backwards compatability
6638 delta = deltaY === 0 ? deltaX : deltaY;
6639
6640 // New school wheel delta (wheel event)
6641 if ( 'deltaY' in orgEvent ) {
6642     deltaY = orgEvent.deltaY * -1;
6643     delta = deltaY;
6644 }
6645 if ( 'deltaX' in orgEvent ) {
6646     deltaX = orgEvent.deltaX;
6647     if ( deltaY === 0 ) { delta = deltaX * -1; }
6648 }
6649
6650 // No change actually happened, no reason to go any further
6651 if ( deltaY === 0 && deltaX === 0 ) { return; }
6652
6653 // Need to convert lines and pages to pixels if we aren't already in pixels
6654 // There are three delta modes:
6655 // * deltaMode 0 is by pixels, nothing to do
6656 // * deltaMode 1 is by lines
6657 // * deltaMode 2 is by pages
6658 if ( orgEvent.deltaMode === 1 ) {
6659     var lineHeight = $data(this, 'mousewheel-line-height');
6660     delta *= lineHeight;
6661     deltaY *= lineHeight;
6662     deltaX *= lineHeight;
6663 } else if ( orgEvent.deltaMode === 2 ) {
6664     var pageHeight = $data(this, 'mousewheel-page-height');
6665     delta *= pageHeight;
6666     deltaY *= pageHeight;
6667     deltaX *= pageHeight;
6668 }
6669
6670 // Store lowest absolute delta to normalize the delta values
6671 absDelta = Math.max( Math.abs(deltaY), Math.abs(deltaX) );
6672
6673 if ( !lowestDelta || absDelta < lowestDelta ) {
6674     lowestDelta = absDelta;
6675
6676     // Adjust older deltas if necessary
6677     if ( shouldAdjustOldDeltas(orgEvent, absDelta) ) {
6678         lowestDelta /= 40;
6679     }
6680 }
6681
6682 // Adjust older deltas if necessary
6683 if ( shouldAdjustOldDeltas(orgEvent, absDelta) ) {
6684     // Divide all the things by 40!
6685     delta /= 40;
6686     deltaX /= 40;
6687     deltaY /= 40;
6688 }
6689
6690 // Get a whole, normalized value for the deltas
6691 delta = Math[delta >= 1 ? 'floor' : 'ceil'](delta / lowestDelta);
6692 deltaX = Math[deltaX >= 1 ? 'floor' : 'ceil'](deltaX / lowestDelta);
6693 deltaY = Math[deltaY >= 1 ? 'floor' : 'ceil'](deltaY / lowestDelta);
6694
6695 // Normalise offsetX and offsetY properties
6696 if ( special.settings.normalizeOffset && this.getBoundingClientRect() ) {
6697     var boundingRect = this.getBoundingClientRect();
6698     offsetX = event.clientX - boundingRect.left;
6699     offsetY = event.clientY - boundingRect.top;
6700 }
6701
6702 // Add information to the event object
6703 event.deltaX = deltaX;
6704 event.deltaY = deltaY;
6705 event.deltaFactor = lowestDelta;
6706 event.offsetX = offsetX;
6707 event.offsetY = offsetY;
6708 // Go ahead and set deltaMode to 0 since we converted to pixels
6709 // Although this is a little odd since we overwrite the deltaX/Y
6710 // properties with normalized deltas.
6711 event.deltaMode = 0;
6712
6713 // Add event and delta to the front of the arguments
6714 args.unshift(event, delta, deltaX, deltaY);
6715
6716 // Clearout lowestDelta after sometime to better
6717 // handle multiple device types that give different
6718 // a different lowestDelta
6719 // Ex: trackpad = 3 and mouse wheel = 120
6720 if ( nullLowestDeltaTimeout ) { clearTimeout(nullLowestDeltaTimeout); }
6721 nullLowestDeltaTimeout = setTimeout(nullLowestDelta, 200);
6722
6723 return ($event.dispatch || $event.handle).apply(this, args);
6724 }
6725
6726 function nullLowestDelta() {
6727     lowestDelta = null;
6728 }
6729
6730 function shouldAdjustOldDeltas(orgEvent, absDelta) {
6731     // If this is an older event and the delta is divisible by 120,
6732     // then we are assuming that the browser is treating this as an
6733     // older mouse wheel event and that we should divide the deltas
6734     // by 40 to try and get a more usable deltaFactor.
6735     // Side note, this actually impacts the reported scroll distance
6736     // in older browsers and can cause scrolling to be slower than native.
6737     // Turn this off by setting $.event.special.mousewheel.settings.adjustOldDeltas to false.
6738     return special.settings.adjustOldDeltas && orgEvent.type === 'mousewheel' && absDelta % 120 === 0;
6739 }
6740

```

```

6741 });
6742
6743 S2.define('jquery.select2',[
6744   'jquery',
6745   'jquery-mousewheel',
6746
6747   './select2/core',
6748   './select2/defaults',
6749   './select2/utils'
6750 ],function ($,_, Select2, Defaults, Utils) {
6751   if ($ fn select2 == null) {
6752     // All methods that should return the element
6753     var thisMethods = ['open','close','destroy'];
6754
6755     $ fn select2 = function (options) {
6756       options = options || {};
6757
6758       if (typeof options === 'object') {
6759         this.each(function () {
6760           var instanceOptions = $ extend true, {}, options;
6761
6762           var instance = new Select2($(this), instanceOptions);
6763         });
6764
6765         return this;
6766       } else if (typeof options === 'string') {
6767         var ret;
6768         var args = Array.prototype.slice.call(arguments, 1);
6769
6770         this.each(function () {
6771           var instance = Utils.GetData(this, 'select2');
6772
6773           if (instance == null && window.console && console.error) {
6774             console.error(
6775               'The select2(\'' + options + '\') method was called on an ' +
6776               'element that is not using Select2.'
6777             );
6778           }
6779
6780           ret = instance[options].apply(instance, args);
6781         });
6782
6783         // Check if we should be returning `this`
6784         if ($ inArray(options, thisMethods) > -1) {
6785           return this;
6786         }
6787
6788         return ret;
6789       } else {
6790         throw new Error('Invalid arguments for Select2: ' + options);
6791       }
6792     };
6793   }
6794
6795   if ($ fn select2 defaults == null) {
6796     $ fn select2 defaults = Defaults;
6797   }
6798
6799   return Select2;
6800 });
6801
6802 // Return the AMD loader configuration so it can be used outside of this file
6803 return {
6804   define: S2.define,
6805   require: S2.require
6806 };
6807 }());
6808
6809 // Autoload the jQuery bindings
6810 // We know that all of the modules exist above this, so we're safe
6811 var select2 = S2.require('jquery.select2');
6812
6813 // Hold the AMD module references on the jQuery function that was just loaded
6814 // This allows Select2 to use the internal loader outside of this file, such
6815 // as in the language files.
6816 jQuery.fn.select2.amd = S2;
6817
6818 // Return the Select2 instance for anyone who is importing it.
6819 return select2;
6820 });

```

**File: ./static/admin/js/vendor/select2/select2.full.min.js**

```
1 /* Select2 4.0.13 | https://github.com/select2/select2/blob/master/LICENSE.md */
2 function(n){"function"==typeof define&&define.amd?define(["jquery"],n):"object"==typeof module&&module.exports?module.exports=function(e){return void 0===t&&(t=e)||"undefined"!==(typeof t)}:n()};
```

**File: ./static/admin/js/vendor/select2/LICENSE.md**

1 The MIT License (MIT)

2

3 Copyright (c) 2012-2017 Kevin Brown, Igor Vaynberg, and Select2 contributors

4

5 Permission is hereby granted, free of charge, to any person obtaining a copy  
6 of this software and associated documentation files (the "Software"), to deal  
7 in the Software without restriction, including without limitation the rights  
8 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
9 copies of the Software, and to permit persons to whom the Software is  
10 furnished to do so, subject to the following conditions:

11

12 The above copyright notice and this permission notice shall be included in  
13 all copies or substantial portions of the Software.

14

15 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
16 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
17 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
18 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
19 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
20 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN  
21 THE SOFTWARE.

**File: ./static/admin/js/vendor/jquery/jquery.js**

```

1  /*!
2   * jQuery JavaScript Library v3.6.0
3   * https://jquery.com/
4   *
5   * Includes Sizzle.js
6   * https://sizzlejs.com/
7   *
8   * Copyright OpenJS Foundation and other contributors
9   * Released under the MIT license
10  * https://jquery.org/license
11  *
12  * Date: 2021-03-02T17:08Z
13  */
14  (function( global, factory ) {
15
16      "use strict";
17
18      if ( typeof module === "object" && typeof module.exports === "object" ) {
19
20          // For CommonJS and CommonJS-like environments where a proper `window`
21          // is present, execute the factory and get jQuery.
22          // For environments that do not have a `window` with a `document`
23          // (such as Node.js), expose a factory as module.exports.
24          // This accutates the need for the creation of a real `window`.
25          // e.g. var jQuery = require("jquery")(window);
26          // See ticket #14549 for more info.
27          module.exports = global.document ?
28              factory( global, true ) :
29              function( w ) {
30                  if ( !w.document ) {
31                      throw new Error( "jQuery requires a window with a document" );

```

```

32     }
33     return factory( w );
34 };
35 } else {
36     factory( global );
37 }
38
39 // Pass this if window is not defined yet
40 }( typeof window !== "undefined" ? window : this, function( window, noGlobal ) {
41
42     // Edge <= 12 - 13+, Firefox <=18 - 45+, IE 10 - 11, Safari 5.1 - 9+, iOS 6 - 9.1
43     // throw exceptions when non-strict code (e.g., ASP.NET 4.5) accesses strict mode
44     // arguments.callee.caller (trac-13335). But as of jQuery 3.0 (2016), strict mode should be common
45     // enough that all such attempts are guarded in a try block.
46     "use strict";
47
48     var arr = [];
49
50     var getProto = Object.getPrototypeOf;
51
52     var slice = arr.slice;
53
54     var flat = arr.flat ? function( array ) {
55         return arr.flat.call( array );
56     } : function( array ) {
57         return arr.concat.apply( [], array );
58     };
59
60
61     var push = arr.push;
62
63     var indexOf = arr.indexOf;
64
65     var class2type = {};
66
67     var toString = class2type.toString;
68
69     var hasOwn = class2type.hasOwnProperty;
70
71     var fnToString = hasOwn.toString;
72
73     var ObjectFunctionString = fnToString.call( Object );
74
75     var support = {};
76
77     var isFunction = function isFunction( obj ) {
78
79         // Support: Chrome <=57, Firefox <=52
80         // In some browsers, typeof returns "function" for HTML <object> elements
81         // (i.e., typeof document.createElement( "object" ) === "function").
82         // We don't want to classify *any* DOM node as a function.
83         // Support: QtWeb <=3.8.5, WebKit <=534.34, wkhtmltopdf tool <=0.12.5
84         // Plus for old WebKit, typeof returns "function" for HTML collections
85         // (e.g., typeof document.getElementsByTagName("div") === "function"). (gh-4756)
86         return typeof obj === "function" && typeof obj.nodeType !== "number" &&
87             typeof obj.item !== "function";
88     };
89
90
91     var isWindow = function isWindow( obj ) {
92         return obj !== null && obj === obj.window;
93     };
94
95
96     var document = window.document;
97
98
99
100     var preservedScriptAttributes = {
101         type: true,
102         src: true,
103         nonce: true,
104         noModule: true
105     };
106
107     function DOMEval( code, node, doc ) {
108         doc = doc || document;
109
110         var i, val;
111         script = doc.createElement( "script" );
112
113         script.text = code;
114         if ( node ) {
115             for ( i in preservedScriptAttributes ) {
116
117                 // Support: Firefox 64+, Edge 18+
118                 // Some browsers don't support the "nonce" property on scripts.
119                 // On the other hand, just using `getAttribute` is not enough as
120                 // the `nonce` attribute is reset to an empty string whenever it
121                 // becomes browsing-context connected.
122                 // See https://github.com/whatwg/html/issues/2369
123                 // See https://html.spec.whatwg.org/#nonce-attributes
124                 // The `node.getAttribute` check was added for the sake of
125                 // `jQuery.globalEval` so that it can fake a nonce-containing node
126                 // via an object.
127                 val = node[ i ] || node.getAttribute && node.getAttribute( i );
128                 if ( val ) {
129                     script.setAttribute( i, val );
130                 }
131             }
132         }
133         doc.head.appendChild( script ); parentNode.removeChild( script );
134     }
135
136
137     function toType( obj ) {
138         if ( obj === null ) {
139             return obj + "";
140         }
141
142         // Support: Android <=2.3 only (functionish RegExp)
143         return typeof obj === "object" || typeof obj === "function" ?
144             class2type[ toString.call( obj ) ] || "object" :
145             typeof obj;
146     }

```



```

147  /* global Symbol */
148  // Defining this global in .eslintrc.json would create a danger of using the global
149  // unguarded in another place, it seems safer to define global only for this module
150
151
152
153  var
154    version = "3.6.0",
155
156    // Define a local copy of jQuery
157    jQuery = function( selector, context ) {
158
159      // The jQuery object is actually just the init constructor 'enhanced'
160      // Need init if jQuery is called (just allow error to be thrown if not included)
161      return new jQuery.fn.init( selector, context );
162    };
163
164    jQuery.fn = jQuery.prototype = {
165
166      // The current version of jQuery being used
167      jquery: version,
168
169      constructor: jQuery,
170
171      // The default length of a jQuery object is 0
172      length: 0,
173
174      toArray: function() {
175        return slice.call( this );
176      },
177
178      // Get the Nth element in the matched element set OR
179      // Get the whole matched element set as a clean array
180      get: function( num ) {
181
182        // Return all the elements in a clean array
183        if ( num == null ) {
184          return slice.call( this );
185        }
186
187        // Return just the one element from the set
188        return num < 0 ? this[ num + this.length ] : this[ num ];
189      },
190
191      // Take an array of elements and push it onto the stack
192      // (returning the new matched element set)
193      pushStack: function( elems ) {
194
195        // Build a new jQuery matched element set
196        var ret = jQuery.merge( this.constructor(), elems );
197
198        // Add the old object onto the stack (as a reference)
199        ret.prevObject = this;
200
201        // Return the newly-formed element set
202        return ret;
203      },
204
205      // Execute a callback for every element in the matched set.
206      each: function( callback ) {
207        return jQuery.each( this, callback );
208      },
209
210      map: function( callback ) {
211        return this.pushStack( jQuery.map( this, function( elem, i ) {
212          return callback.call( elem, i, elem );
213        } ) );
214      },
215
216      slice: function() {
217        return this.pushStack( slice.apply( this, arguments ) );
218      },
219
220      first: function() {
221        return this.eq( 0 );
222      },
223
224      last: function() {
225        return this.eq( -1 );
226      },
227
228      even: function() {
229        return this.pushStack( jQuery.grep( this, function( _elem, i ) {
230          return ( i + 1 ) % 2;
231        } ) );
232      },
233
234      odd: function() {
235        return this.pushStack( jQuery.grep( this, function( _elem, i ) {
236          return i % 2;
237        } ) );
238      },
239
240      eq: function( i ) {
241        var len = this.length;
242        j = +i + ( i < 0 ? len : 0 );
243        return this.pushStack( j >= 0 && j < len ? [ this[ j ] ] : [] );
244      },
245
246      end: function() {
247        return this.prevObject || this.constructor();
248      },
249
250      // For internal use only.
251      // Behaves like an Array's method, not like a jQuery method.
252      push: push,
253      sort: arr.sort,
254      splice: arr.splice
255    };
256
257    jQuery.extend = jQuery.fn.extend = function() {
258      var options, name, src, copy, copyIsArray, clone;
259      target = arguments[ 0 ] || {};
260      i = 1;
261      length = arguments.length;

```

```

262 deep = false;
263
264 // Handle a deep copy situation
265 if ( typeof target === "boolean" ) {
266     deep = target;
267
268     // Skip the boolean and the target
269     target = arguments[ i ] || {};
270     i++;
271 }
272
273 // Handle case when target is a string or something (possible in deep copy)
274 if ( typeof target !== "object" && !isFunction( target ) ) {
275     target = {};
276 }
277
278 // Extend jQuery itself if only one argument is passed
279 if ( i === length ) {
280     target = this;
281     i--;
282 }
283
284 for ( ; i < length; i++ ) {
285
286     // Only deal with non-null/undefined values
287     if ( ( options = arguments[ i ] ) !== null ) {
288
289         // Extend the base object
290         for ( name in options ) {
291             copy = options[ name ];
292
293             // Prevent Object.prototype pollution
294             // Prevent never-ending loop
295             if ( name === "__proto__" || target === copy ) {
296                 continue;
297             }
298
299             // Recurse if we're merging plain objects or arrays
300             if ( deep && copy && ( jQuery.isPlainObject( copy ) ||
301                 ( copyIsArray = Array.isArray( copy ) ) ) ) {
302                 src = target[ name ];
303
304                 // Ensure proper type for the source value
305                 if ( copyIsArray && !Array.isArray( src ) ) {
306                     clone = [];
307                 } else if ( !copyIsArray && jQuery.isPlainObject( src ) ) {
308                     clone = {};
309                 } else {
310                     clone = src;
311                 }
312                 copyIsArray = false;
313
314                 // Never move original objects, clone them
315                 target[ name ] = jQuery.extend( deep, clone, copy );
316
317                 // Don't bring in undefined values
318             } else if ( ( copy !== undefined ) ) {
319                 target[ name ] = copy;
320             }
321         }
322     }
323 }
324
325 // Return the modified object
326 return target;
327 };
328
329 jQuery.extend( {
330
331     // Unique for each copy of jQuery on the page
332     expando: "jQuery" + ( version + Math.random() ).replace( /\D/g, "" ),
333
334     // Assume jQuery is ready without the ready module
335     isReady: true,
336
337     error: function( msg ) {
338         throw new Error( msg );
339     },
340
341     noop: function() {},
342
343     isPlainObject: function( obj ) {
344         var proto, Ctor;
345
346         // Detect obvious negatives
347         // Use toString instead of jQuery.type to catch host objects
348         if ( !obj || !obj.toString.call( obj ) !== "[object Object]" ) {
349             return false;
350         }
351
352         proto = getProto( obj );
353
354         // Objects with no prototype (e.g., `Object.create( null )`) are plain
355         if ( !proto ) {
356             return true;
357         }
358
359         // Objects with prototype are plain iff they were constructed by a global Object function
360         Ctor = hasOwn.call( proto, "constructor" ) && proto.constructor;
361         return typeof Ctor === "function" && fnToString.call( Ctor ) === ObjectFunctionString;
362     },
363
364     isEmptyObject: function( obj ) {
365         var name;
366
367         for ( name in obj ) {
368             return false;
369         }
370         return true;
371     },
372
373     // Evaluates a script in a provided context; falls back to the global one
374     // if not specified.
375     globalEval: function( code, options, doc ) {
376         DOMEval( code, { nonce: options && options.nonce }, doc );
377     }
378 } );

```

```

377 },
378
379 each: function( obj, callback ) {
380     var length, i = 0;
381
382     if ( isArrayLike( obj ) ) {
383         length = obj.length;
384         for ( ; i < length; i++ ) {
385             if ( callback.call( obj[ i ], i, obj[ i ] ) === false ) {
386                 break;
387             }
388         }
389     } else {
390         for ( i in obj ) {
391             if ( callback.call( obj[ i ], i, obj[ i ] ) === false ) {
392                 break;
393             }
394         }
395     }
396
397     return obj;
398 },
399
400 // results is for internal usage only
401 makeArray: function( arr, results ) {
402     var ret = results || [];
403
404     if ( arr != null ) {
405         if ( isArrayLike( Object( arr ) ) ) {
406             jQuery.merge( ret,
407                 typeof arr === "string" ?
408                     [ arr ] : arr
409             );
410         } else {
411             push.call( ret, arr );
412         }
413     }
414
415     return ret;
416 },
417
418 inArray: function( elem, arr, i ) {
419     return arr == null ? -1 : indexOf.call( arr, elem, i );
420 },
421
422 // Support: Android <=4.0 only, PhantomJS 1 only
423 // push.apply(_, arraylike) throws on ancient WebKit
424 merge: function( first, second ) {
425     var len = +second.length,
426         j = 0,
427         i = first.length;
428
429     for ( ; j < len; j++ ) {
430         first[ i++ ] = second[ j ];
431     }
432
433     first.length = i;
434
435     return first;
436 },
437
438 grep: function( elems, callback, invert ) {
439     var callbackInverse,
440         matches = [],
441         i = 0,
442         length = elems.length,
443         callbackExpect = !invert;
444
445     // Go through the array, only saving the items
446     // that pass the validator function
447     for ( ; i < length; i++ ) {
448         callbackInverse = !callback( elems[ i ], i );
449         if ( callbackInverse !== callbackExpect ) {
450             matches.push( elems[ i ] );
451         }
452     }
453
454     return matches;
455 },
456
457 // arg is for internal usage only
458 map: function( elems, callback, arg ) {
459     var length, value,
460         i = 0,
461         ret = [];
462
463     // Go through the array, translating each of the items to their new values
464     if ( isArrayLike( elems ) ) {
465         length = elems.length;
466         for ( ; i < length; i++ ) {
467             value = callback( elems[ i ], i, arg );
468
469             if ( value != null ) {
470                 ret.push( value );
471             }
472         }
473     }
474
475     // Go through every key on the object,
476     } else {
477         for ( i in elems ) {
478             value = callback( elems[ i ], i, arg );
479
480             if ( value != null ) {
481                 ret.push( value );
482             }
483         }
484     }
485
486     // Flatten any nested arrays
487     return flat.call( ret );
488 },
489
490 // A global GUID counter for objects
491 guid: 1,

```

```

492 // jQuery.support is not used in Core but other projects attach their
493 // properties to it so it needs to exist.
494 support: support
495 };
496
497 if ( typeof Symbol === "function" ) {
498   jQuery.fn[ Symbol.iterator ] = arr[ Symbol.iterator ];
499 }
500
501 // Populate the class2type map
502 jQuery.each( "Boolean Number String Function Array Date RegExp Object Error Symbol".split( " " ),
503   function( i, name ) {
504     class2type[ "[object " + name + "]" ] = name.toLowerCase();
505   } );
506
507 function isArrayLike( obj ) {
508
509   // Support: real iOS 8.2 only (not reproducible in simulator)
510   // `in` check used to prevent JIT error (gh-2145)
511   // hasOwn isn't used here due to false negatives
512   // regarding NodeList length in IE
513   var length = !!obj && "length" in obj && obj.length,
514       type = toType( obj );
515
516   if ( isFunction( obj ) || isWindow( obj ) ) {
517     return false;
518   }
519
520   return type === "array" || length === 0 ||
521     typeof length === "number" && length > 0 && ( length - 1 ) in obj;
522 }
523
524 var Sizzle =
525 /*!
526  * Sizzle CSS Selector Engine v2.3.6
527  * https://sizzlejs.com/
528  *
529  * Copyright JS Foundation and other contributors
530  * Released under the MIT license
531  * https://js.foundation/
532  *
533  * Date: 2021-02-16
534  */
535 ( function( window ) {
536   var i,
537       support,
538       Expr,
539       getText,
540       isXML,
541       tokenize,
542       compile,
543       select,
544       outermostContext,
545       sortInput,
546       hasDuplicate,
547
548       // Local document vars
549       setDocument,
550       document,
551       docElem,
552       documentIsHTML,
553       rbuggyQSA,
554       rbuggyMatches,
555       matches,
556       contains,
557
558       // Instance-specific data
559       expando = "sizzle" + 1, * new Date(),
560       preferredDoc = window.document,
561       dirruns = 0,
562       done = 0,
563       classCache = createCache(),
564       tokenCache = createCache(),
565       compilerCache = createCache(),
566       nonnativeSelectorCache = createCache(),
567       sortOrder = function( a, b ) {
568         if ( a === b ) {
569           hasDuplicate = true;
570         }
571         return 0;
572       },
573
574       // Instance methods
575       hasOwn = ( {} ).hasOwnProperty,
576       arr = [],
577       pop = arr.pop,
578       pushNative = arr.push,
579       push = arr.push,
580       slice = arr.slice,
581
582       // Use a stripped-down indexOf as it's faster than native
583       // https://jsperf.com/thor-indexof-vs-for/5
584       indexOf = function( list, elem ) {
585         var i = 0,
586             len = list.length;
587         for ( ; i < len; i++ ) {
588           if ( list[ i ] === elem ) {
589             return i;
590           }
591         }
592         return -1;
593       },
594
595       booleans = "checked|selected|async|autofocus|autoplay|controls|defer|disabled|hidden|" +
596         "ismap|loop|multiple|open|readonly|required|scoped",
597
598       // Regular expressions
599
600       // http://www.w3.org/TR/css3-selectors/#whitespace
601       whitespace = "[\\x20\\t\\r\\n\\f]",
602
603       // https://www.w3.org/TR/css-syntax-3/#ident-token-diagram
604       identifier = "(?![\\da-fa-f]{1,6})" + whitespace +
605         "?[\\w\\r\\n\\f\\j\\w-]([\\w-]?[\\w-])+",
606
607       // Attribute selectors: http://www.w3.org/TR/selectors/#attribute-selectors

```

```

607 attributes = "\\[" + whitespace + "*(\" + identifier + "\")(?:\" + whitespace +
608
609 // Operator (capture 2)
610 "*(\\*^$|!~]?=)" + whitespace +
611
612 // "Attribute values must be CSS identifiers [capture 5]
613 // or strings [capture 3 or capture 4]"
614 "*(?:'((?:\\W|[^\W])*)|\"((?:\\W|[^\W])*)\"|(\" + identifier + \")))" +
615 whitespace + "^\]",
616
617 pseudos = ":(\" + identifier + "\")(?:\\((" +
618
619 // To reduce the number of selectors needing tokenize in the preFilter, prefer arguments:
620 // 1. quoted (capture 3; capture 4 or capture 5)
621 "\"((?:\\W|[^\W])*)\"|\"((?:\\W|[^\W])*)\"|(\" +
622
623 // 2. simple (capture 6)
624 "((?:\\W|[^\W/\\W])" + attributes + ")*)" +
625
626 // 3. anything else (capture 2)
627 ".*" +
628 ")\\|)",
629
630 // Leading and non-escaped trailing whitespace, capturing some non-whitespace characters preceding the latter
631 rwhitespace = new RegExp(whitespace + "+", "g"),
632 rtrim = new RegExp("^(\" + whitespace + "+|((?:^|\\W)(?\\.\\W\\W)*)" +
633 whitespace + "+$", "g"),
634
635 rcomma = new RegExp("^(\" + whitespace + ".*\" + whitespace + ".*" +
636 rcombinators = new RegExp("^(\" + whitespace + ".*[>+~])" + whitespace + ")" + whitespace +
637 ".*" +
638 rdescend = new RegExp(whitespace + "|>"),
639
640 rpseudo = new RegExp(pseudos),
641 ridentifier = new RegExp("^(\" + identifier + "$"),
642
643 matchExpr = {
644 "ID": new RegExp("^(#(\" + identifier + "))",
645 "CLASS": new RegExp("^(\\.(\" + identifier + "))",
646 "TAG": new RegExp("^((\" + identifier + \"|[*])",
647 "ATTR": new RegExp("^(\" + attributes +
648 "PSEUDO": new RegExp("^(\" + pseudos +
649 "CHILD": new RegExp("^(only|first|last|nth|nth-last)-(child|of-type)(?:\\(| +
650 whitespace + "*(even|odd|(((\\+|\\-|\\W)\\d*)n|) + whitespace + "*(?:\\(|\\+|\\-|) +
651 whitespace + "*(\\d+|))" + whitespace + "^\|)", "i"),
652 "bool": new RegExp("^(?:\" + booleans + ")$", "i"),
653
654 // For use in libraries implementing .is()
655 // We use this for POS matching in `select`
656 "needsContext": new RegExp("^(\" + whitespace +
657 ".*[>+~|:|even|odd|eq|gt|lt|nth|first|last)(?:\\(| + whitespace +
658 ".*(?:\\-|\\d)?\\d*)" + whitespace + "^\|)", "i"),
659 },
660
661 rhtml = /HTML$/i,
662 rinputs = /^(?:input|select|textarea|button)$/i,
663 rheader = /^h\d$/i,
664
665 rnative = /^(^|){}\s*{\s*\[native \w/,
666
667 // Easily-parseable/retrievable ID or TAG or CLASS selectors
668 rquickExpr = /^(?:#([\w-]+)|(\w+)|\.([\w-]+))$/i,
669
670 rsibling = /^[+~]/i,
671
672 // CSS escapes
673 // http://www.w3.org/TR/CSS21/syndata.html#escaped-characters
674 runescape = new RegExp("\\W[\\da-fA-F]{1,6}" + whitespace + "?\\\\W([^\r\n\\f])", "g"),
675 funescape = function(escape, nonHex) {
676     var high = "0x" + escape.slice(1) - 0x10000;
677
678     return nonHex ?
679
680 // Strip the backslash prefix from a non-hex escape sequence
681 nonHex :
682
683 // Replace a hexadecimal escape sequence with the encoded Unicode code point
684 // Support: IE <= 11+
685 // For values outside the Basic Multilingual Plane (BMP), manually construct a
686 // surrogate pair
687 high < 0 ?
688     String.fromCharCode(high + 0x10000) :
689     String.fromCharCode(high >> 10 | 0xD800, high & 0x3FFF | 0xDC00);
690 },
691
692 // CSS string/identifier serialization
693 // https://drafts.csswg.org/cssom/#common-serializing-idioms
694 rcssescape = /([\\0-\\x1\\x7f]|^?d)|^-$|([\\0-\\x1\\x7f\\uFFFF\\w-])/g,
695 fcssescape = function(ch, asCodePoint) {
696     if (asCodePoint) {
697
698 // U+0000 NULL becomes U+FFFD REPLACEMENT CHARACTER
699 if (ch === "\\0") {
700     return "\\uFFFD";
701 }
702
703 // Control characters and (dependent upon position) numbers get escaped as code points
704 return ch.slice(0, -1) + "\\u" +
705     ch.charCodeAt(ch.length - 1).toString(16) + " ";
706 }
707
708 // Other potentially-special ASCII characters get backslash-escaped
709 return "\\" + ch;
710 },
711
712 // Used for iframes
713 // See setDocument()
714 // Removing the function wrapper causes a "Permission Denied"
715 // error in IE
716 unloadHandler = function() {
717     setDocument();
718 },
719
720 inDisabledFieldset = addCombinator(
721     function(elem) {

```

```

722     return elem.disabled === true && elem.nodeName.toLowerCase() === "fieldset";
723 },
724 { dir: "parentNode", next: "legend" }
725 );
726
727 // Optimize for push.apply( _, NodeList )
728 try {
729     push.apply(
730         arr = slice.call( preferredDoc.childNodes ),
731         preferredDoc.childNodes
732     );
733
734     // Support: Android<4.0
735     // Detect silently failing push.apply
736     // eslint-disable-next-line no-unused-expressions
737     arr[ preferredDoc.childNodes.length ] = nodeType;
738 } catch ( e ) {
739     push = { apply: arr.length ?
740
741         // Leverage slice if possible
742         function( target, els ) {
743             pushNative.apply( target, slice.call( els ) );
744         };
745
746         // Support: IE<9
747         // Otherwise append directly
748         function( target, els ) {
749             var j = target.length
750                 i = 0;
751
752             // Can't trust NodeList.length
753             while ( ( target[ j++ ] = els[ i++ ] ) ) {}
754             target.length = j - 1;
755         };
756     };
757 }
758
759 function Sizzle( selector, context, results, seed ) {
760     var m, i, elem, nid, match, groups, newSelector,
761         newContext = context && context.ownerDocument,
762
763         // nodeType defaults to 9, since context defaults to document
764         nodeType = context ? context.nodeType : 9,
765
766         results = results || [],
767
768         // Return early from calls with invalid selector or context
769         if ( typeof selector !== "string" || !selector ||
770             nodeType !== 1 && nodeType !== 9 && nodeType !== 11 ) {
771             return results;
772         }
773
774     // Try to shortcut find operations (as opposed to filters) in HTML documents
775     if ( seed ) {
776         setDocument( context );
777         context = context || document;
778
779         if ( documentIsHTML ) {
780
781             // If the selector is sufficiently simple, try using a "getBy*" DOM method
782             // (excepting DocumentFragment context, where the methods don't exist)
783             if ( nodeType !== 11 && ( match = rquickExpr.exec( selector ) ) ) {
784
785                 // ID selector
786                 if ( ( m = match[ 1 ] ) ) {
787                     if ( ( elem = context.getElementById( m ) ) ) {
788
789                         // Document context
790                         if ( nodeType === 9 ) {
791                             if ( ( elem = context.getElementById( m ) ) ) {
792
793                                 // Support: IE, Opera, Webkit
794                                 // TODO: identify versions
795                                 // getElementById can match elements by name instead of ID
796                                 if ( elem.id === m ) {
797                                     results.push( elem );
798                                     return results;
799                                 }
800                             } else {
801                                 return results;
802                             }
803
804                             // Element context
805                         } else {
806
807                             // Support: IE, Opera, Webkit
808                             // TODO: identify versions
809                             // getElementById can match elements by name instead of ID
810                             if ( newContext && ( elem = newContext.getElementById( m ) ) &&
811                                 contains( context, elem ) &&
812                                 elem.id === m ) {
813
814                                 results.push( elem );
815                                 return results;
816                             }
817                         }
818
819                     // Type selector
820                 } else if ( match[ 2 ] ) {
821                     push.apply( results, context.getElementsByTagName( selector ) );
822                     return results;
823
824                     // Class selector
825                 } else if ( ( m = match[ 3 ] ) && support.getElementsByClassName &&
826                     context.getElementsByClassName ) {
827
828                     push.apply( results, context.getElementsByClassName( m ) );
829                     return results;
830                 }
831             }
832
833             // Take advantage of querySelectorAll
834             if ( support.qsa &&
835                 !nonnativeSelectorCache[ selector + " " ] &&
836                 ( !rbuggyQSA || !rbuggyQSA.test( selector ) ) &&

```

```

837
838 // Support: IE 8 only
839 // Exclude object elements
840 ( nodeType !== 1 || context.nodeName.toLowerCase() !== "object" ) {
841
842     newSelector = selector;
843     newContext = context;
844
845     // qSA considers elements outside a scoping root when evaluating child or
846     // descendant combinators, which is not what we want.
847     // In such cases, we work around the behavior by prefixing every selector in the
848     // list with an ID selector referencing the scope context.
849     // The technique has to be used as well when a leading combinator is used
850     // as such selectors are not recognized by querySelectorAll.
851     // Thanks to Andrew Dupont for this technique.
852     if ( nodeType === 1 &&
853         ( rdescend.test( selector ) || rcombinators.test( selector ) ) ) {
854
855         // Expand context for sibling selectors
856         newContext = rsibling.test( selector ) && testContext( context.parentNode ) ||
857             context;
858
859         // We can use :scope instead of the ID hack if the browser
860         // supports it & if we're not changing the context.
861         if ( newContext !== context || !support.scope ) {
862
863             // Capture the context ID, setting it first if necessary
864             if ( !nid = context.getAttribute( "id" ) ) {
865                 nid = nid.replace( rcssescape, fcssescape );
866             } else {
867                 context.setAttribute( "id", ( nid = expando ) );
868             }
869         }
870
871         // Prefix every selector in the list
872         groups = tokenize( selector );
873         i = groups.length;
874         while ( i-- ) {
875             groups[ i ] = ( nid ? "#" + nid : ":scope" ) + " " +
876                 toSelector( groups[ i ] );
877         }
878         newSelector = groups.join( ", " );
879     }
880
881     try {
882         push.apply( results,
883             newContext.querySelectorAll( newSelector ) );
884     };
885     return results;
886 } catch ( qsaError ) {
887     nonnativeSelectorCache( selector, true );
888 } finally {
889     if ( nid === expando ) {
890         context.removeAttribute( "id" );
891     }
892 }
893 }
894 }
895 }
896
897 // All others
898 return select( selector.replace( rtrim, "$1" ), context, results, seed );
899 }
900
901 /**
902  * Create key-value caches of limited size
903  * @returns {function(string, object)} Returns the Object data after storing it on itself with
904  * property name the (space-suffixed) string and (if the cache is larger than Expr.cacheLength)
905  * deleting the oldest entry
906  */
907 function createCache() {
908     var keys = [];
909
910     function cache( key, value ) {
911
912         // Use (key + " ") to avoid collision with native prototype properties (see Issue #157)
913         if ( keys.push( key + " " ) > Expr.cacheLength ) {
914
915             // Only keep the most recent entries
916             delete cache[ keys.shift() ];
917         }
918         return ( cache[ key + " " ] = value );
919     }
920     return cache;
921 }
922
923 /**
924  * Mark a function for special use by Sizzle
925  * @param {Function} fn The function to mark
926  */
927 function markFunction( fn ) {
928     fn[ expando ] = true;
929     return fn;
930 }
931
932 /**
933  * Support testing using an element
934  * @param {Function} fn Passed the created element and returns a boolean result
935  */
936 function assert( fn ) {
937     var el = document.createElement( "fieldset" );
938
939     try {
940         return !!fn( el );
941     } catch ( e ) {
942         return false;
943     } finally {
944
945         // Remove from its parent by default
946         if ( el.parentNode ) {
947             el.parentNode.removeChild( el );
948         }
949     }
950
951     // release memory in IE
952     el = null;

```

```

952     }
953 }
954
955 /**
956  * Adds the same handler for all of the specified attrs
957  * @param {String} attrs Pipe-separated list of attributes
958  * @param {Function} handler The method that will be applied
959  */
960 function addHandler( attrs, handler ) {
961     var arr = attrs.split( "|" ),
962         i = arr.length;
963
964     while ( i-- ) {
965         Expr.attrHandle[ arr[ i ] ] = handler;
966     }
967 }
968
969 /**
970  * Checks document order of two siblings
971  * @param {Element} a
972  * @param {Element} b
973  * @returns {Number} Returns less than 0 if a precedes b, greater than 0 if a follows b
974  */
975 function siblingCheck( a, b ) {
976     var cur = b && a;
977     diff = cur && a.nodeType === 1 && b.nodeType === 1 &&
978         a.sourceIndex - b.sourceIndex;
979
980     // Use IE sourceIndex if available on both nodes
981     if ( diff ) {
982         return diff;
983     }
984
985     // Check if b follows a
986     if ( cur ) {
987         while ( ( cur = cur.nextSibling ) ) {
988             if ( cur === b ) {
989                 return -1;
990             }
991         }
992     }
993
994     return a ? 1 : -1;
995 }
996
997 /**
998  * Returns a function to use in pseudos for input types
999  * @param {String} type
1000  */
1001 function createInputPseudo( type ) {
1002     return function( elem ) {
1003         var name = elem.nodeName.toLowerCase();
1004         return name === "input" && elem.type === type;
1005     };
1006 }
1007
1008 /**
1009  * Returns a function to use in pseudos for buttons
1010  * @param {String} type
1011  */
1012 function createButtonPseudo( type ) {
1013     return function( elem ) {
1014         var name = elem.nodeName.toLowerCase();
1015         return ( name === "input" || name === "button" ) && elem.type === type;
1016     };
1017 }
1018
1019 /**
1020  * Returns a function to use in pseudos for :enabled/:disabled
1021  * @param {Boolean} disabled true for :disabled; false for :enabled
1022  */
1023 function createDisabledPseudo( disabled ) {
1024
1025     // Known :disabled false positives: fieldset[disabled] > legend:nth-of-type(n+2) :can-disable
1026     return function( elem ) {
1027
1028         // Only certain elements can match :enabled or :disabled
1029         // https://html.spec.whatwg.org/multipage/scripting.html#selector-enabled
1030         // https://html.spec.whatwg.org/multipage/scripting.html#selector-disabled
1031         if ( "form" in elem ) {
1032
1033             // Check for inherited disabledness on relevant non-disabled elements:
1034             // * listed form-associated elements in a disabled fieldset
1035             //   https://html.spec.whatwg.org/multipage/forms.html#category-listed
1036             // * option elements in a disabled optgroup
1037             //   https://html.spec.whatwg.org/multipage/forms.html#concept-option-disabled
1038             // All such elements have a "form" property.
1039             if ( elem.parentNode && elem.disabled === false ) {
1040
1041                 // Option elements defer to a parent optgroup if present
1042                 if ( "label" in elem ) {
1043                     if ( "label" in elem.parentNode ) {
1044                         return elem.parentNode.disabled === disabled;
1045                     } else {
1046                         return elem.disabled === disabled;
1047                     }
1048                 }
1049             }
1050
1051             // Support: IE 6 - 11
1052             // Use the isDisabled shortcut property to check for disabled fieldset ancestors
1053             return elem.isDisabled === disabled ||
1054
1055                 // Where there is no isDisabled, check manually
1056                 /* jshint -W018 */
1057                 elem.isDisabled != disabled &&
1058                 inDisabledFieldset( elem ) === disabled;
1059         }
1060
1061         return elem.disabled === disabled;
1062     }
1063
1064     // Try to winnow out elements that can't be disabled before trusting the disabled property.
1065     // Some victims get caught in our net (label, legend, menu, track), but it shouldn't
1066     // even exist on them, let alone have a boolean value.
1067     } else if ( "label" in elem ) {

```



```

1067     return elem.disabled === disabled;
1068 }
1069
1070 // Remaining elements are neither :enabled nor :disabled
1071 return false;
1072 };
1073 }
1074
1075 /**
1076  * Returns a function to use in pseudos for positionals
1077  * @param {Function} fn
1078  */
1079 function createPositionalPseudo( fn ) {
1080     return markFunction( function( argument ) {
1081         argument = +argument;
1082         return markFunction( function( seed, matches ) {
1083             var j;
1084             matchIndexes = fn( [], seed.length, argument );
1085             i = matchIndexes.length;
1086
1087             // Match elements found at the specified indexes
1088             while ( i-- ) {
1089                 if ( seed[ ( j = matchIndexes[ i ] ) ] ) {
1090                     seed[ j ] = !( matches[ j ] = seed[ j ] );
1091                 }
1092             }
1093         } );
1094     } );
1095 }
1096
1097 /**
1098  * Checks a node for validity as a Sizzle context
1099  * @param {Element|Object=} context
1100  * @returns {Element|Object|Boolean} The input node if acceptable, otherwise a falsy value
1101  */
1102 function testContext( context ) {
1103     return context && typeof context.getElementsByTagName !== "undefined" && context;
1104 }
1105
1106 // Expose support vars for convenience
1107 support = Sizzle.support = {};
1108
1109 /**
1110  * Detects XML nodes
1111  * @param {Element|Object} elem An element or a document
1112  * @returns {Boolean} True iff elem is a non-HTML XML node
1113  */
1114 isXML = Sizzle.isXML = function( elem ) {
1115     var namespace = elem && elem.namespaceURI;
1116     docElem = elem && ( elem.ownerDocument || elem ).documentElement;
1117
1118     // Support: IE <= 8
1119     // Assume HTML when documentElement doesn't yet exist, such as inside loading iframes
1120     // https://bugs.jquery.com/ticket/4833
1121     return !rhtml.test( namespace || docElem && docElem.nodeName || "HTML" );
1122 };
1123
1124 /**
1125  * Sets document-related variables once based on the current document
1126  * @param {Element|Object} [doc] An element or document object to use to set the document
1127  * @returns {Object} Returns the current document
1128  */
1129 setDocument = Sizzle.setDocument = function( node ) {
1130     var hasCompare, subWindow;
1131     doc = node ? node.ownerDocument || node : preferredDoc;
1132
1133     // Return early if doc is invalid or already selected
1134     // Support: IE 11+, Edge 17 - 18+
1135     // IE/Edge sometimes throw a "Permission denied" error when strict-comparing
1136     // two documents; shallow comparisons work.
1137     // eslint-disable-next-line eqeqeq
1138     if ( doc === document || doc.nodeType !== 9 || !doc.documentElement ) {
1139         return document;
1140     }
1141
1142     // Update global variables
1143     document = doc;
1144     docElem = document.documentElement;
1145     documentIsHTML = !isXML( document );
1146
1147     // Support: IE 9 - 11+, Edge 12 - 18+
1148     // Accessing iframe documents after unload throws "permission denied" errors (jQuery #13936)
1149     // Support: IE 11+, Edge 17 - 18+
1150     // IE/Edge sometimes throw a "Permission denied" error when strict-comparing
1151     // two documents; shallow comparisons work.
1152     // eslint-disable-next-line eqeqeq
1153     if ( preferredDoc !== document && ( subWindow = document.defaultView ) && subWindow.top !== subWindow ) {
1154         // Support: IE 11, Edge
1155         if ( subWindow.addEventListener ) {
1156             subWindow.addEventListener( "unload", unloadHandler, false );
1157         }
1158         // Support: IE 9 - 10 only
1159         } else if ( subWindow.attachEvent ) {
1160             subWindow.attachEvent( "onunload", unloadHandler );
1161         }
1162     }
1163
1164     // Support: IE 8 - 11+, Edge 12 - 18+, Chrome <=16 - 25 only, Firefox <=3.6 - 31 only,
1165     // Safari 4 - 5 only, Opera <=11.6 - 12.x only
1166     // IE/Edge & older browsers don't support the :scope pseudo-class.
1167     // Support: Safari 6.0 only
1168     // Safari 6.0 supports :scope but it's an alias of :root there.
1169     support.scope = assert( function( el ) {
1170         docElem.appendChild( el ).appendChild( document.createElement( "div" ) );
1171         return typeof el.querySelectorAll !== "undefined" &&
1172             !el.querySelectorAll( ":scope fieldset div" ).length;
1173     } );
1174
1175     // Attributes
1176     // ----- */
1177
1178     // Support: IE < 8
1179     // Verify that getAttribute really returns attributes and not properties

```

```

1182 // (excepting IE8 booleans)
1183 support.attributes = assert( function( el ) {
1184     el.className = "i";
1185     return !el.getAttribute( "className" );
1186 } );
1187
1188 /* getElement(s)By*
1189 ----- */
1190
1191 // Check if getElementsByTagName("**") returns only elements
1192 support.getElementsByTagName = assert( function( el ) {
1193     el.appendChild( document.createComment( "" ) );
1194     return !el.getElementsByTagName( "**" ).length;
1195 } );
1196
1197 // Support: IE<9
1198 support.getElementsByClassName = rnative.test( document.getElementsByClassName );
1199
1200 // Support: IE<10
1201 // Check if getElementById returns elements by name
1202 // The broken getElementById methods don't pick up programmatically-set names,
1203 // so use a roundabout getElementsByName test
1204 support.getById = assert( function( el ) {
1205     docElem.appendChild( el ).id = expando;
1206     return document.getElementsByName( el.id ).length === 1;
1207 } );
1208
1209 // ID filter and find
1210 if ( support.getById ) {
1211     Expr.filter[ "ID" ] = function( id ) {
1212         var attrId = id.replace( runescape, funescape );
1213         return function( elem ) {
1214             return elem.getAttribute( "id" ) === attrId;
1215         };
1216     };
1217     Expr.find[ "ID" ] = function( id, context ) {
1218         if ( typeof context.getElementById !== "undefined" && document.isHTML ) {
1219             var elem = context.getElementById( id );
1220             return elem ? [ elem ] : [];
1221         }
1222     };
1223 } else {
1224     Expr.filter[ "ID" ] = function( id ) {
1225         var attrId = id.replace( runescape, funescape );
1226         return function( elem ) {
1227             var node = typeof elem.getAttributeNode !== "undefined" &&
1228                 elem.getAttributeNode( "id" );
1229             return node && node.value === attrId;
1230         };
1231     };
1232
1233     // Support: IE 6 - 7 only
1234     // getElementById is not reliable as a find shortcut
1235     Expr.find[ "ID" ] = function( id, context ) {
1236         if ( typeof context.getElementById !== "undefined" && document.isHTML ) {
1237             var node, i, elems;
1238             elem = context.getElementById( id );
1239
1240             if ( elem ) {
1241                 // Verify the id attribute
1242                 node = elem.getAttributeNode( "id" );
1243                 if ( node && node.value === id ) {
1244                     return [ elem ];
1245                 }
1246             }
1247
1248             // Fall back on getElementsByName
1249             elems = context.getElementsByName( id );
1250             i = 0;
1251             while ( ( elem = elems[ i++ ] ) ) {
1252                 node = elem.getAttributeNode( "id" );
1253                 if ( node && node.value === id ) {
1254                     return [ elem ];
1255                 }
1256             }
1257         }
1258     };
1259     return [];
1260 }
1261 };
1262 }
1263
1264 // Tag
1265 Expr.find[ "TAG" ] = support.getElementsByTagName ?
1266     function( tag, context ) {
1267         if ( typeof context.getElementsByTagName !== "undefined" ) {
1268             return context.getElementsByTagName( tag );
1269         }
1270     } :
1271     function( tag, context ) {
1272         var fragment = context.appendChild( document.createElement( tag ) );
1273         return fragment.childNodes;
1274     };
1275
1276 function( tag, context ) {
1277     var elem,
1278         tmp = [],
1279         i = 0;
1280
1281     // By happy coincidence, a (broken) gEBTN appears on DocumentFragment nodes too
1282     results = context.getElementsByTagName( tag );
1283
1284     // Filter out possible comments
1285     if ( tag === "*" ) {
1286         while ( ( elem = results[ i++ ] ) ) {
1287             if ( elem.nodeType === 1 ) {
1288                 tmp.push( elem );
1289             }
1290         }
1291     }
1292     return tmp;
1293 }
1294 return results;
1295 };
1296

```

```

2298 // Class
2299 Expr find( "CLASS" ) = support.getElementsByClassName( && function( className, context ) {
2300   if ( typeof context.getElementsByClassName != "undefined" && document.isHTML ) {
2301     return context.getElementsByClassName( className );
2302   }
2303 });
2304
2305 /* QSA/matchesSelector
2306 ----- */
2307
2308 // QSA and matchesSelector support
2309
2310 // matchesSelector(:active) reports false when true (IE9/Opera 11.5)
2311 rbuggyMatches = [];
2312
2313 // qSa(:focus) reports false when true (Chrome 21)
2314 // We allow this because of a bug in IE8/9 that throws an error
2315 // whenever 'document.activeElement' is accessed on an iframe
2316 // So, we allow :focus to pass through QSA all the time to avoid the IE error
2317 // See https://bugs.jquery.com/ticket/13378
2318 rbuggyQSA = [];
2319
2320 if ( !support.qsa || !nativeTest( document.querySelector() ) ) {
2321   // Build QSA regex
2322   // Regex strategy adopted from Diego Perini
2323   assert( function( el ) {
2324     var input
2325
2326     // Select is set to empty string on purpose
2327     // This is to test IE's treatment of not explicitly
2328     // setting a boolean content attribute,
2329     // since its presence should be enough
2330     // https://bugs.jquery.com/ticket/12359
2331     docElem.appendChild( el ).innerHTML = "<a id='\" + expando + \"'></a>" +
2332       "<select id='\" + expando + \"-\\N\\\" msallowcapture='\" +
2333         "\"option selected='\"></option></select>";
2334
2335     // Support: IE8, Opera 11-12.16
2336     // Nothing should be selected when empty strings follow ^= or $= or *=
2337     // The test attribute must be unknown in Opera but "safe" for WinRT
2338     // https://msdn.microsoft.com/en-us/library/ie/hh465388.aspx#attribute_section
2339     if ( el.querySelectorAll( "[msallowcapture^=]" ).length ) {
2340       rbuggyQSA.push( "[*^$]=+" + whitespace + "(?:'|\\\\\\\\)" );
2341     }
2342
2343     // Support: IE8
2344     // Boolean attributes and "value" are not treated correctly
2345     if ( !el.querySelectorAll( ":selected" ).length ) {
2346       rbuggyQSA.push( "\\[" + whitespace + "*(?:value|"+ booleans + ")" );
2347     }
2348
2349     // Support: Chrome<29, Android<4.4, Safari<7.0+, iOS<7.0+, PhantomJS<1.9.8+
2350     if ( !el.querySelectorAll( "[id~=" + expando + "-]" ).length ) {
2351       rbuggyQSA.push( "~=" );
2352     }
2353
2354     // Support: IE 11+, Edge 15 - 18+
2355     // IE 11/Edge don't find elements on a '[name=""]' query in some cases.
2356     // Adding a temporary attribute to the document before the selection works
2357     // around the issue.
2358     // Interestingly, IE 10 & older don't seem to have the issue.
2359     input = document.createElement( "input" );
2360     input.setAttribute( "name", "" );
2361     el.appendChild( input );
2362     if ( !el.querySelectorAll( "[name=\"\"]" ).length ) {
2363       rbuggyQSA.push( "\\[" + whitespace + "name" + whitespace + "*=" +
2364         whitespace + "(?:'|\\\\\\\\)" );
2365     }
2366
2367     // Webkit/Opera - :checked should return selected option elements
2368     // http://www.w3.org/TR/2011/REC-css3-selectors-20110929/#checked
2369     // IE8 throws error here and will not see later tests
2370     if ( !el.querySelectorAll( ":checked" ).length ) {
2371       rbuggyQSA.push( ":checked" );
2372     }
2373
2374     // Support: Safari 8+, iOS 8+
2375     // https://bugs.webkit.org/show_bug.cgi?id=136851
2376     // In-page `selector#id sibling-combinator selector` fails
2377     if ( !el.querySelectorAll( "a#" + expando + "+" ).length ) {
2378       rbuggyQSA.push( ".#,+[+-]" );
2379     }
2380
2381     // Support: Firefox <3.6 - 5 only
2382     // Old Firefox doesn't throw on a badly-escaped identifier.
2383     el.querySelectorAll( "\\W" );
2384     rbuggyQSA.push( "\\W\\W\\W" );
2385   } );
2386
2387   assert( function( el ) {
2388     el.innerHTML = "<a href='\" disabled='disabled'></a>" +
2389       "<select disabled='\"></select>";
2390
2391     // Support: Windows 8 Native Apps
2392     // The type and name attributes are restricted during .innerHTML assignment
2393     var input = document.createElement( "input" );
2394     input.setAttribute( "type", "hidden" );
2395     el.appendChild( input ).setAttribute( "name", "D" );
2396
2397     // Support: IE8
2398     // Enforce case-sensitivity of name attribute
2399     if ( !el.querySelectorAll( "[name=d]" ).length ) {
2400       rbuggyQSA.push( "name" + whitespace + ".*[*$!~]?=" );
2401     }
2402
2403     // FF 3.5 - :enabled/:disabled and hidden elements (hidden elements are still enabled)
2404     // IE8 throws error here and will not see later tests
2405     if ( !el.querySelectorAll( ":enabled" ).length == 2 ) {
2406       rbuggyQSA.push( ":enabled", ":disabled" );
2407     }
2408
2409     // Support: IE9-11+
2410     // IE's :disabled selector does not pick up the children of disabled fieldsets

```

```

1412 docElem.appendChild( el ); disabled = true;
1413 if ( el.querySelectorAll( ":disabled" ).length !== 2 ) {
1414     buggyQSA.push( ":enabled", ":disabled" );
1415 }
1416
1417 // Support: Opera 10 - 11 only
1418 // Opera 10-11 does not throw on post-comma invalid pseudos
1419 el.querySelectorAll( "*,:x" );
1420 buggyQSA.push( "*,:x" );
1421 } );
1422 }
1423
1424 if ( !support.matchesSelector = rnative.test( ( matches = docElem.matches ||
1425     docElem.webkitMatchesSelector ||
1426     docElem.mozMatchesSelector ||
1427     docElem.oMatchesSelector ||
1428     docElem.msMatchesSelector ) ) ) {
1429
1430     assert( function( el ) {
1431
1432         // Check to see if it's possible to do matchesSelector
1433         // on a disconnected node (IE 9)
1434         support.disconnectedMatch = matches.call( el, "*" );
1435
1436         // This should fail with an exception
1437         // Gecko does not error, returns false instead
1438         matches.call( el, "[s!=t];x" );
1439         buggyMatches.push( "[!=" , pseudos );
1440     } );
1441 }
1442
1443 buggyQSA = buggyQSA.length && new RegExp( buggyQSA.join( "|" ) );
1444 buggyMatches = buggyMatches.length && new RegExp( buggyMatches.join( "|" ) );
1445
1446 /* Contains
1447 ----- */
1448 hasCompare = rnative.test( docElem.compareDocumentPosition );
1449
1450 // Element contains another
1451 // Purposefully self-exclusive
1452 // As in, an element does not contain itself
1453 contains = hasCompare || rnative.test( docElem.contains ) ?
1454     function( a, b ) {
1455         var adown = a.nodeType === 9 ? a.documentElement : a,
1456             bup = b && b.parentNode;
1457         return a === bup || !!( bup && bup.nodeType === 1 && (
1458             adown.contains ?
1459                 adown.contains( bup ) :
1460                 a.compareDocumentPosition && a.compareDocumentPosition( bup ) & 16
1461             ) );
1462     } :
1463     function( a, b ) {
1464         if ( b ) {
1465             while ( ( b = b.parentNode ) ) {
1466                 if ( b === a ) {
1467                     return true;
1468                 }
1469             }
1470         }
1471         return false;
1472     };
1473
1474 /* Sorting
1475 ----- */
1476
1477 // Document order sorting
1478 sortOrder = hasCompare ?
1479     function( a, b ) {
1480
1481         // Flag for duplicate removal
1482         if ( a === b ) {
1483             hasDuplicate = true;
1484             return 0;
1485         }
1486
1487         // Sort on method existence if only one input has compareDocumentPosition
1488         var compare = a.compareDocumentPosition - b.compareDocumentPosition;
1489         if ( compare ) {
1490             return compare;
1491         }
1492
1493         // Calculate position if both inputs belong to the same document
1494         // Support: IE 11+, Edge 17 - 18+
1495         // IE/Edge sometimes throw a "Permission denied" error when strict-comparing
1496         // two documents; shallow comparisons work.
1497         // eslint-disable-next-line eqeqeq
1498         compare = ( a.ownerDocument || a ) === ( b.ownerDocument || b ) ?
1499             a.compareDocumentPosition( b ) :
1500
1501         // Otherwise we know they are disconnected
1502         1;
1503
1504         // Disconnected nodes
1505         if ( compare & 1 ||
1506             ( !support.sortDetached && b.compareDocumentPosition( a ) === compare ) ) {
1507
1508             // Choose the first element that is related to our preferred document
1509             // Support: IE 11+, Edge 17 - 18+
1510             // IE/Edge sometimes throw a "Permission denied" error when strict-comparing
1511             // two documents; shallow comparisons work.
1512             // eslint-disable-next-line eqeqeq
1513             if ( a === document || a.ownerDocument === preferredDoc &&
1514                 contains( preferredDoc, a ) ) {
1515                 return -1;
1516             }
1517
1518             // Support: IE 11+, Edge 17 - 18+
1519             // IE/Edge sometimes throw a "Permission denied" error when strict-comparing
1520             // two documents; shallow comparisons work.
1521             // eslint-disable-next-line eqeqeq
1522             if ( b === document || b.ownerDocument === preferredDoc &&
1523                 contains( preferredDoc, b ) ) {
1524                 return 1;
1525             }
1526

```

```

1527 // Maintain original order
1528 return sortInput ?
1529   ( indexOf( sortInput. a ) - indexOf( sortInput. b ) ) :
1530   0;
1531 }
1532
1533 return compare & 4 ? -1 : 1;
1534 };
1535 function( a, b ) {
1536
1537   // Exit early if the nodes are identical
1538   if ( a === b ) {
1539     hasDuplicate = true;
1540     return 0;
1541   }
1542
1543   var cur,
1544       i = 0,
1545       aup = a.parentNode,
1546       bup = b.parentNode,
1547       ap = [ a ],
1548       bp = [ b ];
1549
1550   // Parentless nodes are either documents or disconnected
1551   if ( !aup || !bup ) {
1552
1553     // Support: IE 11+, Edge 17 - 18+
1554     // IE/Edge sometimes throw a "Permission denied" error when strict-comparing
1555     // two documents; shallow comparisons work.
1556     /* eslint-disable eqeqeq */
1557     return a === document ? -1 :
1558           b === document ? 1 :
1559           /* eslint-enable eqeqeq */
1560           aup ? -1 :
1561           bup ? 1 :
1562           sortInput ?
1563             ( indexOf( sortInput. a ) - indexOf( sortInput. b ) ) :
1564             0;
1565
1566     // If the nodes are siblings, we can do a quick check
1567   } else if ( aup === bup ) {
1568     return siblingCheck( a, b );
1569   }
1570
1571   // Otherwise we need full lists of their ancestors for comparison
1572   cur = a;
1573   while ( ( cur = cur.parentNode ) ) {
1574     ap.unshift( cur );
1575   }
1576   cur = b;
1577   while ( ( cur = cur.parentNode ) ) {
1578     bp.unshift( cur );
1579   }
1580
1581   // Walk down the tree looking for a discrepancy
1582   while ( ap[ i ] === bp[ i ] ) {
1583     i++;
1584   }
1585
1586   return i ?
1587
1588     // Do a sibling check if the nodes have a common ancestor
1589     siblingCheck( ap[ i ], bp[ i ] ) :
1590
1591     // Otherwise nodes in our document sort first
1592     // Support: IE 11+, Edge 17 - 18+
1593     // IE/Edge sometimes throw a "Permission denied" error when strict-comparing
1594     // two documents; shallow comparisons work.
1595     /* eslint-disable eqeqeq */
1596     ap[ i ] === preferredDoc ? -1 :
1597     bp[ i ] === preferredDoc ? 1 :
1598     /* eslint-enable eqeqeq */
1599     0;
1600   };
1601
1602   return document;
1603 };
1604
1605 Sizzle.matches = function( expr, elements ) {
1606   return Sizzle( expr, null, null, elements );
1607 };
1608
1609 Sizzle.matchesSelector = function( elem, expr ) {
1610   setDocument( elem );
1611
1612   if ( !support.matchesSelector && document.isHTML &&
1613     !nonnativeSelectorCache[ expr + " " ] &&
1614     ( !rbuggyMatches || !rbuggyMatches.test( expr ) ) &&
1615     ( !rbuggyQSA || !rbuggyQSA.test( expr ) ) ) {
1616
1617     try {
1618       var ret = matches.call( elem, expr );
1619
1620       // IE 9's matchesSelector returns false on disconnected nodes
1621       if ( !ret || !support.disconnectedMatch ||
1622
1623         // As well, disconnected nodes are said to be in a document
1624         // fragment in IE 9
1625         elem.document && elem.document.nodeType !== 11 ) {
1626         return ret;
1627       }
1628     } catch ( e ) {
1629       nonnativeSelectorCache[ expr ] = true;
1630     }
1631   }
1632
1633   return Sizzle( expr, document, null, [ elem ] ).length > 0;
1634 };
1635
1636 Sizzle.contains = function( context, elem ) {
1637
1638   // Set document vars if needed
1639   // Support: IE 11+, Edge 17 - 18+
1640   // IE/Edge sometimes throw a "Permission denied" error when strict-comparing
1641   // two documents; shallow comparisons work.

```

```

1642 // eslint-disable-next-line eqeqeq
1643 if ( ( context.ownerDocument || context ) !== document ) {
1644     setDocument( context );
1645 }
1646 return contains( context, elem );
1647 };
1648
1649 Sizzle.attr = function( elem, name ) {
1650
1651     // Set document vars if needed
1652     // Support: IE 11+, Edge 17 - 18+
1653     // IE/Edge sometimes throw a "Permission denied" error when strict-comparing
1654     // two documents; shallow comparisons work.
1655     // eslint-disable-next-line eqeqeq
1656     if ( ( elem.ownerDocument || elem ) !== document ) {
1657         setDocument( elem );
1658     }
1659
1660     var fn = Expr.attrHandle[ name.toLowerCase() ],
1661         val = fn && hasOwn.call( Expr.attrHandle, name.toLowerCase() ) ?
1662             fn( elem, name, documentIsHTML ) :
1663             undefined;
1664
1665     return val !== undefined ?
1666         val :
1667         support.attributes || !documentIsHTML ?
1668             elem.getAttribute( name ) :
1669             ( val = elem.getAttributeNode( name ) ) && val.specified ?
1670                 val.value :
1671                 null;
1672 };
1673
1674 Sizzle.escape = function( sel ) {
1675     return ( sel + "" ).replace( rcssescape, fcssescape );
1676 };
1677
1678 Sizzle.error = function( msg ) {
1679     throw new Error( "Syntax error, unrecognized expression: " + msg );
1680 };
1681
1682 /**
1683  * Document sorting and removing duplicates
1684  * @param {ArrayLike} results
1685  */
1686 Sizzle.uniqueSort = function( results ) {
1687     var elem,
1688         duplicates = [],
1689         j = 0,
1690         i = 0;
1691
1692     // Unless we *know* we can detect duplicates, assume their presence
1693     hasDuplicate = !support.detectDuplicates;
1694     sortInput = !support.sortStable && results.slice( 0 );
1695     results.sort( sortOrder );
1696
1697     if ( hasDuplicate ) {
1698         while ( ( elem = results[ i++ ] ) ) {
1699             if ( elem === results[ i ] ) {
1700                 j = duplicates.push( i );
1701             }
1702         }
1703         while ( j-- ) {
1704             results.splice( duplicates[ j ], 1 );
1705         }
1706     }
1707
1708     // Clear input after sorting to release objects
1709     // See https://github.com/jquery/sizzle/pull/225
1710     sortInput = null;
1711
1712     return results;
1713 };
1714
1715 /**
1716  * Utility function for retrieving the text value of an array of DOM nodes
1717  * @param {Array|Element} elem
1718  */
1719 getText = Sizzle.getText = function( elem ) {
1720     var node,
1721         ret = "",
1722         i = 0,
1723         nodeType = elem.nodeType;
1724
1725     if ( !nodeType ) {
1726         // If no nodeType, this is expected to be an array
1727         while ( ( node = elem[ i++ ] ) ) {
1728             // Do not traverse comment nodes
1729             ret += getText( node );
1730         }
1731     } else if ( nodeType === 1 || nodeType === 9 || nodeType === 11 ) {
1732         // Use.textContent for elements
1733         // innerText usage removed for consistency of new lines (jQuery #11153)
1734         if ( typeof elem.textContent === "string" ) {
1735             return elem.textContent;
1736         } else {
1737             // Traverse its children
1738             for ( elem = elem.firstChild; elem; elem = elem.nextSibling ) {
1739                 ret += getText( elem );
1740             }
1741         }
1742     } else if ( nodeType === 3 || nodeType === 4 ) {
1743         return elem.nodeValue;
1744     }
1745
1746     // Do not include comment or processing instruction nodes
1747
1748     return ret;
1749 };
1750
1751
1752
1753
1754
1755
1756

```

```

1757 Expr = Sizzle.selectors = {
1758
1759 // Can be adjusted by the user
1760 cacheLength: 50,
1761
1762 createPseudo: markFunction,
1763
1764 match: matchExpr,
1765
1766 attrHandle: {},
1767
1768 find: {},
1769
1770 relative: {
1771 ">": { dir: "parentNode", first: true },
1772 "~": { dir: "parentNode" },
1773 "+": { dir: "previousSibling", first: true },
1774 "~+": { dir: "previousSibling" }
1775 },
1776
1777 preFilter: {
1778 "ATTR": function( match ) {
1779     match[ 1 ] = match[ 1 ].replace( runescape, funescape );
1780
1781     // Move the given value to match[3] whether quoted or unquoted
1782     match[ 3 ] = ( match[ 3 ] || match[ 4 ] ||
1783         match[ 5 ] || "" ).replace( runescape, funescape );
1784
1785     if ( match[ 2 ] === "~=" ) {
1786         match[ 3 ] = " " + match[ 3 ] + " ";
1787     }
1788
1789     return match.slice( 0, 4 );
1790 },
1791
1792 "CHILD": function( match ) {
1793     /* matches from matchExpr["CHILD"]
1794      1 type (only|nth|...)
1795      2 what (child|of-type)
1796      3 argument (even|odd|\d*|\d*n|[+-]?\d+)?[...]
1797      4 xn-component of xn+y argument ([+-]?\d*n)
1798      5 sign of xn-component
1799      6 x of xn-component
1800      7 sign of y-component
1801      8 y of y-component
1802      */
1803     match[ 1 ] = match[ 1 ].toLowerCase();
1804
1805     if ( match[ 1 ].slice( 0, 3 ) === "nth" ) {
1806
1807         // nth-* requires argument
1808         if ( !match[ 3 ] ) {
1809             Sizzle.error( match[ 0 ] );
1810         }
1811
1812         // numeric x and y parameters for Expr.filter.CHILD
1813         // remember that false/true cast respectively to 0/1
1814         match[ 4 ] = +( match[ 4 ] ?
1815             match[ 5 ] + ( match[ 6 ] || 1 ) :
1816             2 * ( match[ 3 ] === "even" || match[ 3 ] === "odd" ) );
1817         match[ 5 ] = +( ( match[ 7 ] + match[ 8 ] ) || match[ 3 ] === "odd" );
1818
1819         // other types prohibit arguments
1820     } else if ( match[ 3 ] ) {
1821         Sizzle.error( match[ 0 ] );
1822     }
1823
1824     return match;
1825 },
1826
1827 "PSEUDO": function( match ) {
1828     var excess,
1829         unquoted = !match[ 6 ] && match[ 2 ];
1830
1831     if ( matchExpr[ "CHILD" ].test( match[ 0 ] ) ) {
1832         return null;
1833     }
1834
1835     // Accept quoted arguments as-is
1836     if ( match[ 3 ] ) {
1837         match[ 2 ] = match[ 4 ] || match[ 5 ] || "";
1838
1839         // Strip excess characters from unquoted arguments
1840     } else if ( unquoted && rpseudo.test( unquoted ) &&
1841         // Get excess from tokenize (recursively)
1842         ( excess = tokenize( unquoted, true ) ) &&
1843         // advance to the next closing parenthesis
1844         ( excess = unquoted.indexOf( ")", unquoted.length - excess ) - unquoted.length ) ) {
1845         // excess is a negative index
1846         match[ 0 ] = match[ 0 ].slice( 0, excess );
1847         match[ 2 ] = unquoted.slice( 0, excess );
1848     }
1849
1850     // Return only captures needed by the pseudo filter method (type and argument)
1851     return match.slice( 0, 3 );
1852 },
1853
1854 filter: {
1855
1856 "TAG": function( nodeNameSelector ) {
1857     var nodeName = nodeNameSelector.replace( runescape, funescape ).toLowerCase();
1858     return nodeNameSelector === "*" ?
1859         function() {
1860             return true;
1861         } :
1862         function( elem ) {
1863             return elem.nodeName && elem.nodeName.toLowerCase() === nodeName;
1864         };
1865 },
1866
1867
1868
1869 },
1870
1871

```

```

1872 "CLASS": function( className ) {
1873     var pattern = classCache[ className + " " ];
1874
1875     return pattern ||
1876     ( pattern = new RegExp( "(^|" + whitespace +
1877     ")" + className + "(" + whitespace + "|$)" ) ) && classCache[
1878     className, function( elem ) {
1879         return pattern.test(
1880             typeof elem.className === "string" && elem.className ||
1881             typeof elem.getAttribute !== "undefined" &&
1882             elem.getAttribute( "class" ) ||
1883             ""
1884         );
1885     } );
1886 },
1887
1888 "ATTR": function( name, operator, check ) {
1889     return function( elem ) {
1890         var result = Sizzle.attr( elem, name );
1891
1892         if ( result == null ) {
1893             return operator === "!=";
1894         }
1895         if ( !operator ) {
1896             return true;
1897         }
1898
1899         result += "";
1900
1901         /* eslint-disable max-len */
1902
1903         return operator === "=" ? result === check :
1904             operator === "!=" ? result !== check :
1905             operator === "^=" ? check && result.indexOf( check ) === 0 :
1906             operator === "*=" ? check && result.indexOf( check ) > -1 :
1907             operator === "$=" ? check && result.slice( -check.length ) === check :
1908             operator === "~=" ? ( " " + result.replace( /\s/g, " " ) + " " ).indexOf( check ) > -1 :
1909             operator === "|=" ? result === check || result.slice( 0, check.length + 1 ) === check + "-":
1910             false;
1911         /* eslint-enable max-len */
1912
1913     };
1914 },
1915
1916 "CHILD": function( type, what, _argument, first, last ) {
1917     var simple = type.slice( 0, 3 ) !== "nth",
1918         forward = type.slice( -4 ) !== "last",
1919         ofType = what !== "of-type";
1920
1921     return first === 1 && last === 0 ?
1922
1923         // Shortcut for :nth-*(n)
1924         function( elem ) {
1925             return !elem.parentNode;
1926         } :
1927
1928         function( elem, _context, xml ) {
1929             var cache, uniqueCache, outerCache, node, nodeIndex, start,
1930                 dir, simple, first, last, previousSibling,
1931                 parent = elem.parentNode,
1932                 name = ofType && elem.nodeName.toLowerCase(),
1933                 useCache = !xml && ofType,
1934                 diff = false;
1935
1936             if ( parent ) {
1937
1938                 // :(first|last|only)-(child|of-type)
1939                 if ( simple ) {
1940                     while ( dir ) {
1941                         node = elem;
1942                         while ( ( node = node[ dir ] ) ) {
1943                             if ( ofType ?
1944                                 node.nodeName.toLowerCase() === name :
1945                                 node.nodeType === 1 ) {
1946                                 return false;
1947                             }
1948                         }
1949                     }
1950
1951                     // Reverse direction for :only- (if we haven't yet done so)
1952                     start = dir = type === "only" && !start && "nextSibling";
1953                 }
1954                 return true;
1955             }
1956
1957             start = [ forward ? parent.firstChild : parent.lastChild ];
1958
1959             // non-xml :nth-child(...) stores cache data on `parent`
1960             if ( forward && useCache ) {
1961
1962                 // Seek `elem` from a previously-cached index
1963
1964                 // ...in a gzip-friendly way
1965                 node = parent;
1966                 outerCache = node[ expando ] || ( node[ expando ] = {} );
1967
1968                 // Support: IE <9 only
1969                 // Defend against cloned attrroperties (jQuery gh-1709)
1970                 uniqueCache = outerCache[ node.uniqueID ] ||
1971                 ( outerCache[ node.uniqueID ] = {} );
1972
1973                 cache = uniqueCache[ type ] || [];
1974                 nodeIndex = cache[ 0 ] !== dirruns && cache[ 1 ];
1975                 diff = nodeIndex && cache[ 2 ];
1976                 node = nodeIndex && parent.childNodes[ nodeIndex ];
1977
1978                 while ( ( node = ++nodeIndex && node && node[ dir ] ) ) {
1979
1980                     // Fallback to seeking `elem` from the start
1981                     ( diff = nodeIndex = 0 ) || start.pop() ) {
1982
1983                         // When found, cache indexes on `parent` and break
1984                         if ( node.nodeType === 1 && ++diff && node === elem ) {
1985                             uniqueCache[ type ] = [ dirruns, nodeIndex, diff ];
1986                             break;

```



```

1987     }
1988   }
1989 } else {
1990
1991   // Use previously-cached element index if available
1992   if ( useCache ) {
1993
1994     // ...in a gzip-friendly way
1995     node = elem;
1996     outerCache = node[ expando ] || ( node[ expando ] = {} );
1997
1998     // Support: IE <9 only
1999     // Defend against cloned attrproperties (jQuery gh-1709)
2000     uniqueCache = outerCache[ node.uniqueID ] ||
2001       ( outerCache[ node.uniqueID ] = {} );
2002
2003     cache = uniqueCache[ type ] || [];
2004     nodeIndex = cache[ 0 ] === dirruns && cache[ 1 ];
2005     diff = nodeIndex;
2006   }
2007
2008   // xml :nth-child(...)
2009   // or :nth-last-child(...) or :nth(-last)?-of-type(...)
2010   if ( diff === false ) {
2011
2012     // Use the same loop as above to seek `elem` from the start
2013     while ( ( node = ++nodeIndex && node && node.dir ) ) {
2014       ( diff = nodeIndex = 0 ) || start.pop() ) {
2015
2016         if ( ( ofType ?
2017           node.nodeName.toLowerCase() === name :
2018           node.nodeType === 1 ) &&
2019           ++diff ) {
2020
2021           // Cache the index of each encountered element
2022           if ( useCache ) {
2023             outerCache = node[ expando ] ||
2024               ( node[ expando ] = {} );
2025
2026             // Support: IE <9 only
2027             // Defend against cloned attrproperties (jQuery gh-1709)
2028             uniqueCache = outerCache[ node.uniqueID ] ||
2029               ( outerCache[ node.uniqueID ] = {} );
2030
2031             uniqueCache[ type ] = [ dirruns, diff ];
2032           }
2033
2034           if ( node === elem ) {
2035             break;
2036           }
2037         }
2038       }
2039     }
2040   }
2041 }
2042
2043 // Incorporate the offset, then check against cycle size
2044 diff -= last;
2045 return diff === first || ( diff % first === 0 && diff / first >= 0 );
2046 }
2047 };
2048 },
2049
2050 "PSEUDO": function( pseudo, argument ) {
2051
2052   // pseudo-class names are case-insensitive
2053   // http://www.w3.org/TR/selectors/#pseudo-classes
2054   // Prioritize by case sensitivity in case custom pseudos are added with uppercase letters
2055   // Remember that setFilters inherits from pseudos
2056   var args;
2057   fn = Expr.pseudos[ pseudo ] || Expr.setFilters[ pseudo.toLowerCase() ] ||
2058     Sizzle.error( "unsupported pseudo: " + pseudo );
2059
2060   // The user may use createPseudo to indicate that
2061   // arguments are needed to create the filter function
2062   // just as Sizzle does
2063   if ( fn[ expando ] ) {
2064     return fn( argument );
2065   }
2066
2067   // But maintain support for old signatures
2068   if ( fn.length > 1 ) {
2069     args = [ pseudo, pseudo, "" , argument ];
2070     return Expr.setFilters.hasOwnProperty( pseudo.toLowerCase() ) ?
2071       markFunction( function( seed, matches ) {
2072         var idx,
2073           matched = fn( seed, argument ),
2074           i = matched.length;
2075         while ( i-- ) {
2076           idx = indexOf( seed, matched[ i ] );
2077           seed[ idx ] = !( matches[ idx ] = matched[ i ] );
2078         }
2079       } ) :
2080       function( elem ) {
2081         return fn( elem, 0, args );
2082       };
2083   }
2084
2085   return fn;
2086 }
2087 },
2088
2089 pseudos: {
2090
2091   // Potentially complex pseudos
2092   "not": markFunction( function( selector ) {
2093
2094     // Trim the selector passed to compile
2095     // to avoid treating leading and trailing
2096     // spaces as combinators
2097     var input = [],
2098       results = [],
2099       matcher = compile( selector.replace( rtrim, "$1" ) );
2100
2101     return matcher[ expando ] ?

```

```

2102 markFunction( function( seed, matches, _context, xml ) {
2103     var elem;
2104     unmatched = matcher( seed, null, xml, [] ),
2105     i = seed.length;
2106
2107     // Match elements unmatched by `matcher`
2108     while ( i-- ) {
2109         if ( ( elem = unmatched[ i ] ) ) {
2110             seed[ i ] = !!( matches[ i ] = elem );
2111         }
2112     }
2113     } );
2114     function( elem, _context, xml ) {
2115         input[ 0 ] = elem;
2116         matcher( input, null, xml, results );
2117
2118         // Don't keep the element (issue #299)
2119         input[ 0 ] = null;
2120         return !results.pop();
2121     };
2122 }, ),
2123
2124 "has": markFunction( function( selector ) {
2125     return function( elem ) {
2126         return Sizzle( selector, elem ).length > 0;
2127     };
2128 }, ),
2129
2130 "contains": markFunction( function( text ) {
2131     text = text.replace( runescape, funescape );
2132     return function( elem ) {
2133         return ( elem.textContent || getText( elem ) ).indexOf( text ) > -1;
2134     };
2135 }, ),
2136
2137 // "Whether an element is represented by a :lang() selector
2138 // is based solely on the element's language value
2139 // being equal to the identifier C,
2140 // or beginning with the identifier C immediately followed by "-".
2141 // The matching of C against the element's language value is performed case-insensitively.
2142 // The identifier C does not have to be a valid language name."
2143 // http://www.w3.org/TR/selectors/#lang-pseudo
2144 "lang": markFunction( function( lang ) {
2145
2146     // lang value must be a valid identifier
2147     if ( !identifier.test( lang || "" ) ) {
2148         Sizzle.error( "unsupported lang: " + lang );
2149     }
2150     lang = lang.replace( runescape, funescape ).toLowerCase();
2151     return function( elem ) {
2152         var elemLang;
2153         do {
2154             if ( ( elemLang = documentIsHTML ?
2155                 elem.lang :
2156                 elem.getAttribute( "xml:lang" ) || elem.getAttribute( "lang" ) ) ) {
2157
2158                 elemLang = elemLang.toLowerCase();
2159                 return elemLang === lang || elemLang.indexOf( lang + "-" ) === 0;
2160             }
2161         } while ( ( elem = elem.parentNode ) && elem.nodeType === 1 );
2162         return false;
2163     };
2164 }, ),
2165
2166 // Miscellaneous
2167 "target": function( elem ) {
2168     var hash = window.location && window.location.hash;
2169     return hash && hash.slice( 1 ) === elem.id;
2170 },
2171
2172 "root": function( elem ) {
2173     return elem === docElem;
2174 },
2175
2176 "focus": function( elem ) {
2177     return elem === document.activeElement &&
2178         ( !document.hasFocus() || document.hasFocus() ) &&
2179         !( elem.type || elem.href || -elem.tabIndex );
2180 },
2181
2182 // Boolean properties
2183 "enabled": createDisabledPseudo( false ),
2184 "disabled": createDisabledPseudo( true ),
2185
2186 "checked": function( elem ) {
2187
2188     // In CSS3, :checked should return both checked and selected elements
2189     // http://www.w3.org/TR/2011/REC-css3-selectors-20110929/#checked
2190     var nodeName = elem.nodeName.toLowerCase();
2191     return ( nodeName === "input" && !elem.checked ) ||
2192         ( nodeName === "option" && !elem.selected );
2193 },
2194
2195 "selected": function( elem ) {
2196
2197     // Accessing this property makes selected-by-default
2198     // options in Safari work properly
2199     if ( elem.parentNode ) {
2200         // eslint-disable-next-line no-unused-expressions
2201         elem.parentNode.selectedIndex;
2202     }
2203
2204     return elem.selected === true;
2205 },
2206
2207 // Contents
2208 "empty": function( elem ) {
2209
2210     // http://www.w3.org/TR/selectors/#empty-pseudo
2211     // :empty is negated by element (1) or content nodes (text: 3; cdata: 4; entity ref: 5),
2212     // but not by others (comment: 8; processing instruction: 7; etc.)
2213     // nodeName < 6 works because attributes (2) do not appear as children
2214     for ( elem = elem.firstChild; elem = elem.nextSibling; ) {
2215         if ( elem.nodeType < 6 ) {
2216             return false;

```

```

2217     }
2218   }
2219   return true;
2220 },
2221
2222 "parent": function( elem ) {
2223   return 'Expr pseudos[ "empty" ]'( elem );
2224 },
2225
2226 // Element/input types
2227 "header": function( elem ) {
2228   return rheader.test( elem.nodeName );
2229 },
2230
2231 "input": function( elem ) {
2232   return rinputs.test( elem.nodeName );
2233 },
2234
2235 "button": function( elem ) {
2236   var name = elem.nodeName.toLowerCase();
2237   return name === "input" && elem.type === "button" || name === "button";
2238 },
2239
2240 "text": function( elem ) {
2241   var attr;
2242   return elem.nodeName.toLowerCase() === "input" &&
2243     elem.type === "text" &&
2244
2245     // Support: IE<8
2246     // New HTML5 attribute values (e.g., "search") appear with elem.type === "text"
2247     ( ( attr = elem.getAttribute( "type" ) ) == null ||
2248       attr.toLowerCase() === "text" );
2249 },
2250
2251 // Position-in-collection
2252 "first": createPositionalPseudo( function() {
2253   return [ 0 ];
2254 } ),
2255
2256 "last": createPositionalPseudo( function( _matchIndexes, length ) {
2257   return [ length - 1 ];
2258 } ),
2259
2260 "eq": createPositionalPseudo( function( _matchIndexes, length, argument ) {
2261   return [ argument < 0 ? argument + length : argument ];
2262 } ),
2263
2264 "even": createPositionalPseudo( function( matchIndexes, length ) {
2265   var i = 0;
2266   for ( ; i < length; i += 2 ) {
2267     matchIndexes.push( i );
2268   }
2269   return matchIndexes;
2270 } ),
2271
2272 "odd": createPositionalPseudo( function( matchIndexes, length ) {
2273   var i = 1;
2274   for ( ; i < length; i += 2 ) {
2275     matchIndexes.push( i );
2276   }
2277   return matchIndexes;
2278 } ),
2279
2280 "lt": createPositionalPseudo( function( matchIndexes, length, argument ) {
2281   var i = argument < 0 ?
2282     argument + length :
2283     argument > length ?
2284       length :
2285       argument;
2286   for ( ; -i >= 0; ) {
2287     matchIndexes.push( i );
2288   }
2289   return matchIndexes;
2290 } ),
2291
2292 "gt": createPositionalPseudo( function( matchIndexes, length, argument ) {
2293   var i = argument < 0 ? argument + length : argument;
2294   for ( ; ++i < length; ) {
2295     matchIndexes.push( i );
2296   }
2297   return matchIndexes;
2298 } )
2299 }
2300 };
2301
2302 Expr.pseudos[ "nth" ] = Expr.pseudos[ "eq" ];
2303
2304 // Add button/input type pseudos
2305 for ( i in { radio: true, checkbox: true, file: true, password: true, image: true } ) {
2306   Expr.pseudos[ i ] = createInputPseudo( i );
2307 }
2308 for ( i in { submit: true, reset: true } ) {
2309   Expr.pseudos[ i ] = createButtonPseudo( i );
2310 }
2311
2312 // Easy API for creating new setFilters
2313 function setFilters() {}
2314 setFilters.prototype = Expr.filters = Expr.pseudos;
2315 Expr.setFilters = new setFilters();
2316
2317 tokenize = Sizzle.tokenize = function( selector, parseOnly ) {
2318   var matched, match, tokens, type;
2319   soFar, groups, preFilters;
2320   cached = tokenCache[ selector + " " ];
2321
2322   if ( cached ) {
2323     return parseOnly ? 0 : cached.slice( 0 );
2324   }
2325
2326   soFar = selector;
2327   groups = [];
2328   preFilters = Expr.preFilter;
2329
2330   while ( soFar ) {

```

```

2332 // Comma and first run
2333 if ( !matched || ( match = rcomma.exec( soFar ) ) ) {
2334   if ( match ) {
2335     // Don't consume trailing commas as valid
2336     soFar = soFar.slice( match[ 0 ].length ) || soFar;
2337   }
2338   groups.push( { tokens: [] } );
2339 }
2340
2341 matched = false;
2342
2343 // Combinators
2344 if ( ( match = rcombinators.exec( soFar ) ) ) {
2345   matched = match.shift();
2346   tokens.push( {
2347     value: matched,
2348   } );
2349   // Cast descendant combinators to space
2350   type: match[ 0 ].replace( rtrim, " " );
2351 } );
2352 soFar = soFar.slice( matched.length );
2353 }
2354
2355 // Filters
2356 for ( type in Expr.filter ) {
2357   if ( ( match = matchExpr[ type ].exec( soFar ) ) && ( !preFilters[ type ] ||
2358     ( match = preFilters[ type ].match( match ) ) ) ) {
2359     matched = match.shift();
2360     tokens.push( {
2361       value: matched,
2362       type: type,
2363       matches: match
2364     } );
2365     soFar = soFar.slice( matched.length );
2366   }
2367 }
2368
2369 if ( !matched ) {
2370   break;
2371 }
2372 }
2373 }
2374
2375 // Return the length of the invalid excess
2376 // If we're just parsing
2377 // Otherwise, throw an error or return tokens
2378 return parseOnly ?
2379   soFar.length :
2380   soFar ?
2381     Sizzle.error( selector ) :
2382     // Cache the tokens
2383     tokenCache[ selector ].groups.slice( 0 );
2384 };
2385
2386 function toSelector( tokens ) {
2387   var i = 0,
2388       len = tokens.length,
2389       selector = "";
2390   for ( ; i < len; i++ ) {
2391     selector += tokens[ i ].value;
2392   }
2393   return selector;
2394 }
2395
2396 function addCombinator( matcher, combinator, base ) {
2397   var dir = combinator.dir,
2398       skip = combinator.next,
2399       key = skip || dir,
2400       checkNonElements = base && key === "parentNode",
2401       doneName = done++;
2402   return combinator.first ?
2403     // Check against closest ancestor/preceding element
2404     function( elem, context, xml ) {
2405       while ( ( elem = elem[ dir ] ) ) {
2406         if ( elem.nodeType === 1 || checkNonElements ) {
2407           return matcher( elem, context, xml );
2408         }
2409       }
2410       return false;
2411     } :
2412     // Check against all ancestor/preceding elements
2413     function( elem, context, xml ) {
2414       var oldCache = uniqueCache, outerCache,
2415           newCache = [ dirruns, doneName ];
2416       // We can't set arbitrary data on XML nodes, so they don't benefit from combinator caching
2417       if ( xml ) {
2418         while ( ( elem = elem[ dir ] ) ) {
2419           if ( elem.nodeType === 1 || checkNonElements ) {
2420             if ( !matcher( elem, context, xml ) ) {
2421               return true;
2422             }
2423           }
2424         }
2425       } else {
2426         if ( elem.nodeType === 1 || checkNonElements ) {
2427           outerCache = elem[ expando ] || ( elem[ expando ] = {} );
2428           if ( !oldCache ) {
2429             uniqueCache = { dirruns: 0, doneName: "" };
2430             outerCache[ uniqueCache.dirruns ] = uniqueCache;
2431             outerCache[ uniqueCache.doneName ] = uniqueCache;
2432           }
2433           if ( !oldCache[ doneName ] ) {
2434             outerCache[ doneName ] = true;
2435           }
2436         }
2437       }
2438       return !oldCache[ doneName ];
2439     }
2440 }
2441
2442 // Support: IE <9 only
2443 // Defend against cloned attrroperties (jQuery gh-1709)
2444 uniqueCache = { dirruns: 0, doneName: "" };
2445 ( uniqueCache[ uniqueCache.dirruns ] = uniqueCache );
2446
2447 if ( !skip && skip === elem.nodeName.toLowerCase() ) {
2448   elem = elem[ dir ] || elem;
2449 } else if ( !oldCache[ key ] &&
2450   oldCache[ 0 ] === dirruns && oldCache[ 1 ] === doneName ) {
2451   // Assign to newCache so results back-propagate to previous elements
2452   return ( newCache[ 2 ] = oldCache[ 2 ] );
2453 }

```

```

2447     } else {
2448
2449         // Reuse newcache so results back-propagate to previous elements
2450         uniqueCache[ key ] = newCache;
2451
2452         // A match means we're done; a fail means we have to keep checking
2453         if ( ( newCache[ 2 ] = matcher( elem, context, xml ) ) ) {
2454             return true;
2455         }
2456     }
2457 }
2458 }
2459 }
2460 return false;
2461 };
2462 }
2463
2464 function elementMatcher( matchers ) {
2465     return matchers.length > 1 ?
2466         function( elem, context, xml ) {
2467             var i = matchers.length;
2468             while ( i-- ) {
2469                 if ( !matchers[ i ]( elem, context, xml ) ) {
2470                     return false;
2471                 }
2472             }
2473             return true;
2474         } :
2475         matchers[ 0 ];
2476 }
2477
2478 function multipleContexts( selector, contexts, results ) {
2479     var i = 0,
2480         len = contexts.length;
2481     for ( ; i < len; i++ ) {
2482         Sizzle( selector, contexts[ i ], results );
2483     }
2484     return results;
2485 }
2486
2487 function condense( unmatched, map, filter, context, xml ) {
2488     var elem,
2489         newUnmatched = [],
2490         i = 0,
2491         len = unmatched.length,
2492         mapped = map != null;
2493     for ( ; i < len; i++ ) {
2494         if ( ( elem = unmatched[ i ] ) ) {
2495             if ( !filter || filter( elem, context, xml ) ) {
2496                 newUnmatched.push( elem );
2497                 if ( mapped ) {
2498                     map.push( i );
2499                 }
2500             }
2501         }
2502     }
2503     return newUnmatched;
2504 }
2505
2506 function setMatcher( preFilter, selector, matcher, postFilter, postFinder, postSelector ) {
2507     if ( !postFilter && !postFinder && !postSelector ) {
2508         postFilter = setMatcher( postFilter );
2509     }
2510     if ( !postFinder && !postFinder && !postSelector ) {
2511         postFinder = setMatcher( postFinder, postSelector );
2512     }
2513     return markFunction( function( seed, results, context, xml ) {
2514         var temp, i, elem,
2515             preMap = [],
2516             postMap = [],
2517             preexisting = results.length;
2518
2519         // Get initial elements from seed or context
2520         elems = seed || multipleContexts(
2521             selector, context.nodeType ? [ context ] : context,
2522             []
2523         ),
2524
2525         // Prefilter to get matcher input, preserving a map for seed-results synchronization
2526         matcherIn = preFilter && ( seed || !selector ) ?
2527             condense( elems, preMap, preFilter, context, xml ) :
2528             elems,
2529
2530         matcherOut = matcher ?
2531             // If we have a postFinder, or filtered seed, or non-seed postFilter or preexisting results,
2532             postFinder || ( seed ? preFilter : preexisting || postFilter ) ?
2533                 // ...intermediate processing is necessary
2534                 [] :
2535                 // ...otherwise use results directly
2536                 results :
2537             matcherIn;
2538
2539         // Find primary matches
2540         if ( !matcher ) {
2541             matcher( matcherIn, matcherOut, context, xml );
2542         }
2543
2544         // Apply postFilter
2545         if ( postFilter ) {
2546             temp = [];
2547             for ( i = 0; i < matcherOut.length; i++ ) {
2548                 if ( postFilter( matcherOut[ i ], i, context, xml ) ) {
2549                     temp.push( matcherOut[ i ] );
2550                 }
2551             }
2552             matcherOut = temp;
2553         }
2554
2555         // Un-match failing elements by moving them back to matcherIn
2556         i = 0;
2557         while ( i < matcherOut.length ) {
2558             if ( !matcherIn[ i ] ) {
2559                 matcherOut[ i ] = null;
2560             }
2561         }
2562     } );

```

```

2562 }
2563
2564 if ( seed ) {
2565   if ( postFinder || preFilter ) {
2566     if ( postFinder ) {
2567       // Get the final matcherOut by condensing this intermediate into postFinder contexts
2568       temp = [];
2569       i = matcherOut.length;
2570       while ( i-- ) {
2571         if ( ( elem = matcherOut[ i ] ) ) {
2572           // Restore matcherIn since elem is not yet a final match
2573           temp.push( ( matcherIn[ i ] = elem ) );
2574         }
2575       }
2576       postFinder( null, ( matcherOut = [] ), temp, xml );
2577     }
2578     // Move matched elements from seed to results to keep them synchronized
2579     i = matcherOut.length;
2580     while ( i-- ) {
2581       if ( ( elem = matcherOut[ i ] ) &&
2582         ( temp = postFinder ? indexOf( seed, elem ) : preMap( i ) > -1 ) ) {
2583         seed[ temp ] = !( results[ temp ] = elem );
2584       }
2585     }
2586   }
2587   // Add elements to results, through postFinder if defined
2588   } else {
2589     matcherOut = condense(
2590       matcherOut === results ?
2591         matcherOut.splice( preexisting, matcherOut.length ) :
2592         matcherOut
2593     );
2594     if ( postFinder ) {
2595       postFinder( null, results, matcherOut, xml );
2596     } else {
2597       push.apply( results, matcherOut );
2598     }
2599   }
2600 }
2601
2602 function matcherFromTokens( tokens ) {
2603   var checkContext, matcher, j,
2604   len = tokens.length,
2605   leadingRelative = Expr.relative[ tokens[ 0 ].type ],
2606   implicitRelative = leadingRelative || Expr.relative[ " " ],
2607   i = leadingRelative ? 1 : 0,
2608
2609   // The foundational matcher ensures that elements are reachable from top-level context(s)
2610   matchContext = addCombinator( function( elem ) {
2611     return elem === checkContext;
2612   }, implicitRelative, true ),
2613   matchAnyContext = addCombinator( function( elem ) {
2614     return indexOf( checkContext, elem ) > -1;
2615   }, implicitRelative, true ),
2616   matchers = [ function( elem, context, xml ) {
2617     var ret = ( !leadingRelative && ( xml || context ) !== outermostContext ) || (
2618       ( checkContext = context ).nodeType ?
2619         matchContext( elem, context, xml ) :
2620         matchAnyContext( elem, context, xml ) );
2621   } ];
2622
2623   // Avoid hanging onto element (issue #299)
2624   checkContext = null;
2625   return ret;
2626 }
2627
2628 for ( ; i < len; i++ ) {
2629   if ( ( matcher = Expr.relative[ tokens[ i ].type ] ) ) {
2630     matchers = [ addCombinator( elementMatcher( matchers ), matcher ) ];
2631   } else {
2632     matcher = Expr.filter[ tokens[ i ].type ].apply( null, tokens[ i ].matches );
2633
2634     // Return special upon seeing a positional matcher
2635     if ( matcher.expando ) {
2636       // Find the next relative operator (if any) for proper handling
2637       j = ++i;
2638       for ( ; j < len; j++ ) {
2639         if ( Expr.relative[ tokens[ j ].type ] ) {
2640           break;
2641         }
2642       }
2643       return setMatcher(
2644         i > 1 && elementMatcher( matchers ),
2645         i > 1 && toSelector(
2646           // If the preceding token was a descendant combinator, insert an implicit any-element '*'
2647           tokens
2648             .slice( 0, i - 1 )
2649             .concat( { value: tokens[ i - 2 ].type === " " ? "*" : "" } )
2650             .replace( rtrim, "$1" ),
2651           matcher
2652         ),
2653         j < len && matcherFromTokens( tokens.slice( i, j ) ),
2654         j < len && toSelector( tokens )
2655       );
2656     }
2657     matchers.push( matcher );
2658   }
2659 }
2660
2661 return elementMatcher( matchers );
2662 }
2663
2664 function matcherFromGroupMatchers( elementMatchers, setMatchers ) {
2665   var bySet = setMatchers.length > 0,
2666   byElement = elementMatchers.length > 0,
2667   superMatcher = function( seed, context, xml, results, outermost ) {
2668     var elem, j, matcher,
2669     matchedCount = 0;

```

```

2677 i = "0",
2678 unmatched = seed && [],
2679 setMatched = [],
2680 contextBackup = outermostContext;
2681
2682 // We must always have either seed elements or outermost context
2683 elems = seed || byElement && Expr.find( "TAG" )( "**", outermost );
2684
2685 // Use integer dirruns iff this is the outermost matcher
2686 dirrunsUnique = ( dirruns += contextBackup == null ? 1 : Math.random() || 0.1 ),
2687 len = elems.length;
2688
2689 if ( outermost ) {
2690
2691     // Support: IE 11+, Edge 17 - 18+
2692     // IE/Edge sometimes throw a "Permission denied" error when strict-comparing
2693     // two documents; shallow comparisons work.
2694     // eslint-disable-next-line eqeqeq
2695     outermostContext = context == document || context || outermost;
2696 }
2697
2698 // Add elements passing elementMatchers directly to results
2699 // Support: IE<9, Safari
2700 // Tolerate NodeList properties (IE: "length"; Safari: <number>) matching elements by id
2701 for ( ; i !== len && ( elem = elems[ i ] ) != null; i++ ) {
2702     if ( byElement && elem ) {
2703         j = 0;
2704
2705         // Support: IE 11+, Edge 17 - 18+
2706         // IE/Edge sometimes throw a "Permission denied" error when strict-comparing
2707         // two documents; shallow comparisons work.
2708         // eslint-disable-next-line eqeqeq
2709         if ( !context && elem.ownerDocument !== document ) {
2710             setDocument( elem );
2711             xml = !documentIsHTML;
2712         }
2713         while ( ( matcher = elementMatchers[ j++ ] ) ) {
2714             if ( matcher( elem, context || document, xml ) ) {
2715                 results.push( elem );
2716                 break;
2717             }
2718         }
2719         if ( outermost ) {
2720             dirruns = dirrunsUnique;
2721         }
2722     }
2723
2724     // Track unmatched elements for set filters
2725     if ( bySet ) {
2726
2727         // They will have gone through all possible matchers
2728         if ( ( elem = 'matcher' && elem ) ) {
2729             matchedCount--;
2730         }
2731
2732         // Lengthen the array for every element, matched or not
2733         if ( seed ) {
2734             unmatched.push( elem );
2735         }
2736     }
2737 }
2738
2739 // `i` is now the count of elements visited above, and adding it to `matchedCount`
2740 // makes the latter nonnegative.
2741 matchedCount += i;
2742
2743 // Apply set filters to unmatched elements
2744 // NOTE: This can be skipped if there are no unmatched elements (i.e., `matchedCount`
2745 // equals `i`), unless we didn't visit _any_ elements in the above loop because we have
2746 // no element matchers and no seed.
2747 // Incrementing an initially-string "0" `i` allows `i` to remain a string only in that
2748 // case, which will result in a "00" `matchedCount` that differs from `i` but is also
2749 // numerically zero.
2750 if ( bySet && i !== matchedCount ) {
2751     j = 0;
2752     while ( ( matcher = setMatchers[ j++ ] ) ) {
2753         matcher( unmatched, setMatched, context, xml );
2754     }
2755
2756     if ( seed ) {
2757
2758         // Reintegrate element matches to eliminate the need for sorting
2759         if ( matchedCount > 0 ) {
2760             while ( i-- ) {
2761                 if ( !unmatched[ i ] || !setMatched[ i ] ) {
2762                     setMatched[ i ] = pop.call( results );
2763                 }
2764             }
2765         }
2766
2767         // Discard index placeholder values to get only actual matches
2768         setMatched = condense( setMatched );
2769     }
2770
2771     // Add matches to results
2772     push.apply( results, setMatched );
2773
2774     // Seedless set matches succeeding multiple successful matchers stipulate sorting
2775     if ( outermost && !seed && setMatched.length > 0 &&
2776         ( matchedCount + setMatchers.length ) > 1 ) {
2777
2778         Sizzle.uniqueSort( results );
2779     }
2780 }
2781
2782 // Override manipulation of globals by nested matchers
2783 if ( outermost ) {
2784     dirruns = dirrunsUnique;
2785     outermostContext = contextBackup;
2786 }
2787
2788 return unmatched;
2789 };
2790
2791 return bySet ?

```

```

2792 markFunction( superMatcher );
2793 superMatcher:
2794 }
2795
2796 compile = Sizzle.compile = function( selector, match /* Internal Use Only */ ) {
2797     var i;
2798     setMatchers = [];
2799     elementMatchers = [];
2800     cached = compilerCache[ selector + " " ];
2801
2802     if ( !cached ) {
2803
2804         // Generate a function of recursive functions that can be used to check each element
2805         if ( !match ) {
2806             match = tokenize( selector );
2807         }
2808         i = match.length;
2809         while ( i-- ) {
2810             cached = matcherFromTokens( match[ i ] );
2811             if ( cached[ expando ] ) {
2812                 setMatchers.push( cached );
2813             } else {
2814                 elementMatchers.push( cached );
2815             }
2816         }
2817
2818         // Cache the compiled function
2819         cached = compilerCache[
2820             selector,
2821             matcherFromGroupMatchers( elementMatchers, setMatchers )
2822         ];
2823
2824         // Save selector and tokenization
2825         cached.selector = selector;
2826     }
2827     return cached;
2828 };
2829
2830 /**
2831  * A low-level selection function that works with Sizzle's compiled
2832  * selector functions
2833  * @param {String|Function} selector A selector or a pre-compiled
2834  * selector function built with Sizzle.compile
2835  * @param {Element} context
2836  * @param {Array} [results]
2837  * @param {Array} [seed] A set of elements to match against
2838  */
2839 select = Sizzle.select = function( selector, context, results, seed ) {
2840     var i, tokens, token, type, find,
2841         compiled = typeof selector === "function" && selector,
2842         match = !seed && tokenize( selector = compiled.selector || selector );
2843
2844     results = results || [];
2845
2846     // Try to minimize operations if there is only one selector in the list and no seed
2847     // (the latter of which guarantees us context)
2848     if ( match.length === 1 ) {
2849
2850         // Reduce context if the leading compound selector is an ID
2851         tokens = match[ 0 ] = match[ 0 ].slice( 0 );
2852         if ( tokens.length > 2 && ( token = tokens[ 0 ] ).type === "ID" &&
2853             context.nodeType === 9 && documentIsHTML && Expr.relative[ tokens[ 1 ].type ] ) {
2854
2855             context = ( Expr.find[ "ID" ]( token.matches[ 0 ]
2856                 .replace( runescape, funescape ), context ) )[ 0 ];
2857             if ( !context ) {
2858                 return results;
2859             }
2860
2861             // Precompiled matchers will still verify ancestry, so step up a level
2862             context = context.parentNode;
2863         }
2864
2865         selector = selector.slice( tokens.shift().value.length );
2866     }
2867
2868     // Fetch a seed set for right-to-left matching
2869     i = matchExpr[ "needsContext" ].test( selector ) ? 0 : tokens.length;
2870     while ( i-- ) {
2871         token = tokens[ i ];
2872
2873         // Abort if we hit a combinator
2874         if ( Expr.relative[ ( type = token.type ) ] ) {
2875             break;
2876         }
2877         if ( ( find = Expr.find[ type ] ) ) {
2878
2879             // Search, expanding context for leading sibling combinators
2880             if ( ( seed = find(
2881                 token.matches[ 0 ].replace( runescape, funescape ),
2882                 rsibling.test( tokens[ 0 ].type ) && testContext( context.parentNode ) ||
2883                 context
2884             ) ) ) {
2885
2886                 // If seed is empty or no tokens remain, we can return early
2887                 tokens.splice( i, 1 );
2888                 selector = seed.length && toSelector( tokens );
2889                 if ( !selector ) {
2890                     push.apply( results, seed );
2891                     return results;
2892                 }
2893
2894                 break;
2895             }
2896         }
2897     }
2898
2899     // Compile and execute a filtering function if one is not provided
2900     // Provide `match` to avoid retokenization if we modified the selector above
2901     ( compiled || compile( selector, match ) )(
2902         seed,
2903         context,
2904         documentIsHTML,
2905         results
2906     );

```



```

2907     context || rsibling.test( selector ) && testContext( context.parentNode ) || context
2908 };
2909 return results;
2910 };
2911
2912 // One-time assignments
2913
2914 // Sort stability
2915 support.sortStable = expando.split( "" ).sort( sortOrder ).join( "" ) === expando;
2916
2917 // Support: Chrome 14-35+
2918 // Always assume duplicates if they aren't passed to the comparison function
2919 support.detectDuplicates = !hasDuplicate;
2920
2921 // Initialize against the default document
2922 setDocument();
2923
2924 // Support: Webkit<537.32 - Safari 6.0.3/Chrome 25 (fixed in Chrome 27)
2925 // Detached nodes confoundingly follow *each other*
2926 support.sortDetached = assert( function( el ) {
2927
2928     // Should return 1, but returns 4 (following)
2929     return el.compareDocumentPosition( document.createElement( "fieldset" ) ) & 1;
2930 } );
2931
2932 // Support: IE<8
2933 // Prevent attribute/property "interpolation"
2934 // https://msdn.microsoft.com/en-us/library/ms536429%28VS.85%29.aspx
2935 if ( !assert( function( el ) {
2936     el.innerHTML = "<a href='#'></a>";
2937     return el.firstChild.getAttribute( "href" ) === "#";
2938 } ) ) {
2939     addHandle( "type|href|height|width", function( elem, name, isXML ) {
2940         if ( !isXML ) {
2941             return elem.getAttribute( name, name.toLowerCase() ) === "type" ? 1 : 2;
2942         }
2943     } );
2944 }
2945
2946 // Support: IE<9
2947 // Use defaultValue in place of getAttribute("value")
2948 if ( !support.attributes || !assert( function( el ) {
2949     el.innerHTML = "<input/>";
2950     el.firstChild.setAttribute( "value", "" );
2951     return el.firstChild.getAttribute( "value" ) === "";
2952 } ) ) {
2953     addHandle( "value", function( elem, _name, isXML ) {
2954         if ( !isXML && elem.nodeName.toLowerCase() === "input" ) {
2955             return elem.defaultValue;
2956         }
2957     } );
2958 }
2959
2960 // Support: IE<9
2961 // Use getAttributeNode to fetch booleans when getAttribute lies
2962 if ( !assert( function( el ) {
2963     return el.getAttribute( "disabled" ) == null;
2964 } ) ) {
2965     addHandle( "booleans", function( elem, name, isXML ) {
2966         var val;
2967         if ( !isXML ) {
2968             return elem[ name ] === true ? name.toLowerCase() :
2969                 ( val = elem.getAttributeNode( name ) ) && val.specified ?
2970                     val.value :
2971                     null;
2972         }
2973     } );
2974 }
2975
2976 return Sizzle;
2977
2978 } )( window );
2979
2980
2981
2982 jQuery.find = Sizzle;
2983 jQuery.expr = Sizzle.selectors;
2984
2985 // Deprecated
2986 jQuery.expr[ ":" ] = jQuery.expr.pseudos;
2987 jQuery.uniqueSort = jQuery.unique = Sizzle.uniqueSort;
2988 jQuery.text = Sizzle.getText;
2989 jQuery.isXMLDoc = Sizzle.isXML;
2990 jQuery.contains = Sizzle.contains;
2991 jQuery.escapeSelector = Sizzle.escape;
2992
2993
2994
2995
2996 var dir = function( elem, dir, until ) {
2997     var matched = [],
2998         truncate = until !== undefined;
2999
3000     while ( ( { elem: elem, dir: dir } ) && elem.nodeType !== 9 ) {
3001         if ( elem.nodeType === 1 ) {
3002             if ( truncate && jQuery( elem ).is( until ) ) {
3003                 break;
3004             }
3005             matched.push( elem );
3006         }
3007     }
3008     return matched;
3009 };
3010
3011
3012 var siblings = function( n, elem ) {
3013     var matched = [];
3014
3015     for ( ; n = n.nextSibling ; ) {
3016         if ( n.nodeType === 1 && n !== elem ) {
3017             matched.push( n );
3018         }
3019     }
3020
3021     return matched;

```

```

3022 };
3023
3024
3025 var rneedsContext = jQuery.expr.match.needsContext;
3026
3027
3028
3029 function nodeName( elem, name ) {
3030
3031     return elem.nodeName.toLowerCase() === name.toLowerCase();
3032 }
3033
3034 var rsingleTag = ( /^<([a-z]|^[\0-\x20\t\r\nf]*)[\x20\t\r\nf]*\?>(?:<\/[1-9]|$)/i );
3035
3036
3037
3038 // Implement the identical functionality for filter and not
3039 function winnow( elements, qualifier, not ) {
3040     if ( isFunction( qualifier ) ) {
3041         return jQuery.grep( elements, function( elem, i ) {
3042             return !qualifier.call( elem, i, elem ) !== not;
3043         } );
3044     }
3045
3046     // Single element
3047     if ( qualifier.nodeType ) {
3048         return jQuery.grep( elements, function( elem ) {
3049             return ( elem === qualifier ) !== not;
3050         } );
3051     }
3052
3053     // Arraylike of elements (jQuery, arguments, Array)
3054     if ( typeof qualifier !== "string" ) {
3055         return jQuery.grep( elements, function( elem ) {
3056             return ( indexOf.call( qualifier, elem ) > -1 ) !== not;
3057         } );
3058     }
3059
3060     // Filtered directly for both simple and complex selectors
3061     return jQuery.filter( qualifier, elements, not );
3062 }
3063
3064 jQuery.filter = function( expr, elems, not ) {
3065     var elem = elems[ 0 ];
3066
3067     if ( not ) {
3068         expr = ":not(" + expr + ")";
3069     }
3070
3071     if ( elems.length === 1 && elem.nodeType === 1 ) {
3072         return jQuery.find.matchesSelector( elem, expr ) ? [ elem ] : [];
3073     }
3074
3075     return jQuery.find.matches( expr, jQuery.grep( elems, function( elem ) {
3076         return elem.nodeType === 1;
3077     } ) );
3078 };
3079
3080 jQuery.fn.extend( {
3081     find: function( selector ) {
3082         var i, ret;
3083         len = this.length;
3084         self = this;
3085
3086         if ( typeof selector !== "string" ) {
3087             return this.pushStack( jQuery( selector ).filter( function() {
3088                 for ( i = 0; i < len; i++ ) {
3089                     if ( jQuery.contains( self[ i ], this ) ) {
3090                         return true;
3091                     }
3092                 }
3093             } ) );
3094         }
3095
3096         ret = this.pushStack( [] );
3097
3098         for ( i = 0; i < len; i++ ) {
3099             jQuery.find( selector, self[ i ], ret );
3100         }
3101
3102         return len > 1 ? jQuery.uniqueSort( ret ) : ret;
3103     },
3104     filter: function( selector ) {
3105         return this.pushStack( winnow( this, selector, || [], false ) );
3106     },
3107     not: function( selector ) {
3108         return this.pushStack( winnow( this, selector, || [], true ) );
3109     },
3110     is: function( selector ) {
3111         return !winnow(
3112             this,
3113             selector,
3114             // If this is a positional/relative selector, check membership in the returned set
3115             // so $("p:first").is("p:last") won't return true for a doc with two "p".
3116             typeof selector === "string" && rneedsContext.test( selector ) ?
3117                 jQuery( selector ) :
3118                 selector || [],
3119             false
3120         ).length;
3121     }
3122 };
3123
3124
3125 // Initialize a jQuery object
3126
3127
3128 // A central reference to the root jQuery(document)
3129 var rootjQuery,
3130
3131 // A simple way to check for HTML strings
3132 // Prioritize #id over <tag> to avoid XSS via location.hash (#9521)
3133 // Strict HTML recognition (#11290: must start with <)
3134 // Shortcut simple #id case for speed
3135 rquickExpr = /^(?:\s*(<[\w\W]+>)|^#[^\s]*|[^\s]*|([^\s\W]+))$/;
3136

```

```

3137 init = jQuery.fn.init = function( selector, context, root ) {
3138     var match, elem;
3139
3140     // HANDLE: $(""), $(null), $(undefined), $(false)
3141     if ( !selector ) {
3142         return this;
3143     }
3144
3145     // Method init() accepts an alternate rootjQuery
3146     // so migrate can support jQuery.sub (gh-2101)
3147     root = root || rootjQuery;
3148
3149     // Handle HTML strings
3150     if ( typeof selector === "string" ) {
3151         if ( selector[ 0 ] === "<" &&
3152             selector[ selector.length - 1 ] === ">" &&
3153             selector.length >= 3 ) {
3154
3155             // Assume that strings that start and end with <> are HTML and skip the regex check
3156             match = [ null, selector, null ];
3157
3158         } else {
3159             match = rquickExpr.exec( selector );
3160         }
3161
3162         // Match html or make sure no context is specified for #id
3163         if ( match && ( match[ 1 ] || !context ) ) {
3164
3165             // HANDLE: $(html) -> $(array)
3166             if ( match[ 1 ] ) {
3167                 context = context instanceof jQuery ? context[ 0 ] : context;
3168
3169                 // Option to run scripts is true for back-compat
3170                 // Intentionally let the error be thrown if parseHTML is not present
3171                 jQuery.merge( this, jQuery.parseHTML(
3172                     match[ 1 ],
3173                     context && context.nodeType ? context.ownerDocument || context : document,
3174                     true
3175                 ) );
3176
3177                 // HANDLE: $(html, props)
3178                 if ( !rsingleTag.test( match[ 1 ] ) && jQuery.isPlainObject( context ) ) {
3179                     for ( match in context ) {
3180
3181                         // Properties of context are called as methods if possible
3182                         if ( isFunction( this[ match ] ) ) {
3183                             this[ match ]( context[ match ] );
3184                         }
3185
3186                         // ...and otherwise set as attributes
3187                         } else {
3188                             this.attr( match, context[ match ] );
3189                         }
3190                     }
3191
3192                     return this;
3193                 }
3194
3195                 // HANDLE: $(#id)
3196                 } else {
3197                     elem = document.getElementById( match[ 2 ] );
3198
3199                     if ( elem ) {
3200                         // Inject the element directly into the jQuery object
3201                         this[ 0 ] = elem;
3202                         this.length = 1;
3203                     }
3204                     return this;
3205                 }
3206
3207                 // HANDLE: $(expr, $(...))
3208             } else if ( !context || context.jquery ) {
3209                 return ( context || root ).find( selector );
3210             }
3211
3212             // HANDLE: $(expr, context)
3213             // (which is just equivalent to: $(context).find(expr)
3214             } else {
3215                 return this.constructor( context ).find( selector );
3216             }
3217
3218             // HANDLE: $(DOMElement)
3219             } else if ( selector.nodeType ) {
3220                 this[ 0 ] = selector;
3221                 this.length = 1;
3222                 return this;
3223             }
3224
3225             // HANDLE: $(function)
3226             // Shortcut for document ready
3227             } else if ( isFunction( selector ) ) {
3228                 return root.ready !== undefined ?
3229                     root.ready( selector ) :
3230
3231                     // Execute immediately if ready is not present
3232                     selector( jQuery );
3233             }
3234
3235             return jQuery.makeArray( selector, this );
3236         };
3237
3238         // Give the init function the jQuery prototype for later instantiation
3239         init.prototype = jQuery.fn;
3240
3241         // Initialize central reference
3242         rootjQuery = jQuery( document );
3243
3244         var rparentsprev = /^(?:parents|prev(?:Until|All))/,
3245
3246         // Methods guaranteed to produce a unique set when starting from a unique set
3247         guaranteedUnique = {
3248             children: true,
3249             contents: true,
3250             next: true,
3251             prev: true
3252         };

```

```

3252
3253 jQuery.fn.extend( {
3254   has: function( target ) {
3255     var targets = jQuery( target, this ),
3256     l = targets.length,
3257
3258     return this.filter( function() {
3259       var i = 0;
3260       for ( ; i < l; i++ ) {
3261         if ( jQuery.contains( this, targets[ i ] ) ) {
3262           return true;
3263         }
3264       }
3265     } );
3266   },
3267
3268   closest: function( selectors, context ) {
3269     var cur,
3270     i = 0,
3271     l = this.length,
3272     matched = [],
3273     targets = typeof selectors !== "string" && jQuery( selectors );
3274
3275     // Positional selectors never match, since there's no _selection_ context
3276     if ( !rneedsContext.test( selectors ) ) {
3277       for ( ; i < l; i++ ) {
3278         for ( cur = this[ i ]; cur && cur !== context; cur = cur.parentNode ) {
3279
3280           // Always skip document fragments
3281           if ( cur.nodeType < 11 && ( targets ?
3282             targets.index( cur ) > -1 :
3283
3284             // Don't pass non-elements to Sizzle
3285             cur.nodeType === 1 &&
3286             jQuery.find.matchesSelector( cur, selectors ) ) ) {
3287
3288             matched.push( cur );
3289             break;
3290           }
3291         }
3292       }
3293     }
3294
3295     return this.pushStack( matched.length > 1 ? jQuery.uniqueSort( matched ) : matched );
3296   },
3297
3298   // Determine the position of an element within the set
3299   index: function( elem ) {
3300
3301     // No argument, return index in parent
3302     if ( !elem ) {
3303       return ( this[ 0 ] && this[ 0 ].parentNode ) ? this.first().prevAll().length : -1;
3304     }
3305
3306     // Index in selector
3307     if ( typeof elem === "string" ) {
3308       return indexOf.call( jQuery( elem ), this[ 0 ] );
3309     }
3310
3311     // Locate the position of the desired element
3312     return indexOf.call( this,
3313
3314       // If it receives a jQuery object, the first element is used
3315       elem.jquery ? elem[ 0 ] : elem
3316     );
3317   },
3318
3319   add: function( selector, context ) {
3320     return this.pushStack(
3321       jQuery.uniqueSort(
3322         jQuery.merge( this.get(), jQuery( selector, context ) )
3323       )
3324     );
3325   },
3326
3327   addBack: function( selector ) {
3328     return this.add( selector === null ?
3329       this.prevObject : this.prevObject.filter( selector )
3330     );
3331   },
3332   };
3333
3334   function sibling( cur, dir ) {
3335     while ( ( cur = cur[ dir ] ) && cur.nodeType !== 1 ) {}
3336     return cur;
3337   }
3338
3339   jQuery.each( {
3340     parent: function( elem ) {
3341       var parent = elem.parentNode;
3342       return parent && parent.nodeType !== 11 ? parent : null;
3343     },
3344     parents: function( elem ) {
3345       return dir( elem, "parentNode" );
3346     },
3347     parentsUntil: function( elem, _i, until ) {
3348       return dir( elem, "parentNode", until );
3349     },
3350     next: function( elem ) {
3351       return sibling( elem, "nextSibling" );
3352     },
3353     prev: function( elem ) {
3354       return sibling( elem, "previousSibling" );
3355     },
3356     nextAll: function( elem ) {
3357       return dir( elem, "nextSibling" );
3358     },
3359     prevAll: function( elem ) {
3360       return dir( elem, "previousSibling" );
3361     },
3362     nextUntil: function( elem, _i, until ) {
3363       return dir( elem, "nextSibling", until );
3364     },
3365     prevUntil: function( elem, _i, until ) {
3366       return dir( elem, "previousSibling", until );

```

```

3367 },
3368 siblings: function( elem ) {
3369     return siblings( ( elem.parentNode || {} ) firstChild elem );
3370 },
3371 children: function( elem ) {
3372     return siblings( elem.firstChild );
3373 },
3374 contents: function( elem ) {
3375     if ( elem.contentDocument != null &&
3376         // Support: IE 11+
3377         // <object> elements with no `data` attribute has an object
3378         // `contentDocument` with a `null` prototype.
3379         getProto( elem.contentDocument ) ) {
3380
3381         return elem.contentDocument;
3382     }
3383 }
3384
3385 // Support: IE 9 - 11 only, iOS 7 only, Android Browser <=4.3 only
3386 // Treat the template element as a regular one in browsers that
3387 // don't support it.
3388 if ( nodeName( elem, "template" ) ) {
3389     elem = elem.content || elem;
3390 }
3391
3392 return jQuery.merge( [], elem.childNodes );
3393 }
3394 }, function( name, fn ) {
3395     jQuery.fn[ name ] = function( until, selector ) {
3396         var matched = jQuery.map( this, fn, until );
3397
3398         if ( name.slice( -5 ) !== "Until" ) {
3399             selector = until;
3400         }
3401
3402         if ( selector && typeof selector === "string" ) {
3403             matched = jQuery.filter( selector, matched );
3404         }
3405
3406         if ( this.length > 1 ) {
3407             // Remove duplicates
3408             if ( !guaranteedUnique[ name ] ) {
3409                 jQuery.uniqueSort( matched );
3410             }
3411
3412             // Reverse order for parents* and prev-derivatives
3413             if ( rparentsprev.test( name ) ) {
3414                 matched.reverse();
3415             }
3416         }
3417
3418         return this.pushStack( matched );
3419     };
3420 } );
3421
3422 var rnothtmlwhite = ( /[^\x20\t\r\n\f]+/g );
3423
3424
3425 // Convert String-formatted options into Object-formatted ones
3426 function createOptions( options ) {
3427     var object = {};
3428     jQuery.each( options.match( rnothtmlwhite ) || [], function( _, flag ) {
3429         object[ flag ] = true;
3430     } );
3431     return object;
3432 }
3433
3434 /*
3435  * Create a callback list using the following parameters:
3436  *
3437  * options: an optional list of space-separated options that will change how
3438  * the callback list behaves or a more traditional option object
3439  *
3440  * By default a callback list will act like an event callback list and can be
3441  * "fired" multiple times.
3442  *
3443  * Possible options:
3444  *
3445  * once: will ensure the callback list can only be fired once (like a Deferred)
3446  *
3447  * memory: will keep track of previous values and will call any callback added
3448  * after the list has been fired right away with the latest "memorized"
3449  * values (like a Deferred)
3450  *
3451  * unique: will ensure a callback can only be added once (no duplicate in the list)
3452  *
3453  * stopOnFalse: interrupt callings when a callback returns false
3454  *
3455  */
3456 jQuery.Callbacks = function( options ) {
3457     // Convert options from String-formatted to Object-formatted if needed
3458     // (we check in cache first)
3459     options = typeof options === "string" ?
3460         createOptions( options ) :
3461         options || {};
3462     jQuery.extend( {}, options );
3463
3464     var // Flag to know if list is currently firing
3465         firing,
3466         // Last fire value for non-forgettable lists
3467         memory,
3468         // Flag to know if list was already fired
3469         fired,
3470         // Flag to prevent firing
3471         locked,
3472         // Actual callback list
3473         list = [],
3474         // Queue of execution data for repeatable lists
3475         queue = [],

```

```

3482
3483 // Index of currently firing callback (modified by add/remove as needed)
3484 firingIndex = -1;
3485
3486 // Fire callbacks
3487 fire = function() {
3488
3489 // Enforce single-firing
3490 locked = locked || options.once;
3491
3492 // Execute callbacks for all pending executions,
3493 // respecting firingIndex overrides and runtime changes
3494 fired = firing = true;
3495 for ( ; queue.length; firingIndex = -1 ) {
3496   memory = queue.shift();
3497   while ( ++firingIndex < list.length ) {
3498
3499 // Run callback and check for early termination
3500 if ( !list[ firingIndex ] || apply( memory, 0 ), memory[ 1 ] ) === false &&
3501   options.stopOnFalse ) {
3502
3503 // Jump to end and forget the data so .add doesn't re-fire
3504 firingIndex = list.length;
3505 memory = false;
3506 }
3507 }
3508 }
3509
3510 // Forget the data if we're done with it
3511 if ( !options.memory ) {
3512   memory = false;
3513 }
3514
3515 firing = false;
3516
3517 // Clean up if we're done firing for good
3518 if ( !locked ) {
3519
3520 // Keep an empty list if we have data for future add calls
3521 if ( memory ) {
3522   list = [];
3523 }
3524 // Otherwise, this object is spent
3525 } else {
3526   list = "";
3527 }
3528 },
3529 },
3530
3531 // Actual Callbacks object
3532 self = {
3533
3534 // Add a callback or a collection of callbacks to the list
3535 add: function() {
3536   if ( !list ) {
3537
3538 // If we have memory from a past run, we should fire after adding
3539 if ( memory && firing ) {
3540   firingIndex = list.length - 1;
3541   queue.push( memory );
3542 }
3543
3544 ( function add( args ) {
3545   jQuery.each( args, function( _, arg ) {
3546     if ( !isFunction( arg ) ) {
3547       if ( !options.unique || !self.has( arg ) ) {
3548         list.push( arg );
3549       }
3550     } else if ( arg && arg.length && toType( arg ) !== "string" ) {
3551
3552 // Inspect recursively
3553       add( arg );
3554     }
3555   } );
3556 } )( arguments );
3557
3558 if ( memory && firing ) {
3559   fire();
3560 }
3561 }
3562 return this;
3563 },
3564
3565 // Remove a callback from the list
3566 remove: function() {
3567   jQuery.each( arguments, function( _, arg ) {
3568     var index;
3569     while ( ( index = jQuery.inArray( arg, list, index ) ) > -1 ) {
3570       list.splice( index, 1 );
3571     }
3572
3573 // Handle firing indexes
3574 if ( index <= firingIndex ) {
3575       firingIndex--;
3576     }
3577   } );
3578   return this;
3579 },
3580
3581 // Check if a given callback is in the list.
3582 // If no argument is given, return whether or not list has callbacks attached.
3583 has: function( fn ) {
3584   return fn ?
3585     jQuery.inArray( fn, list ) > -1 :
3586     list.length > 0;
3587 },
3588
3589 // Remove all callbacks from the list
3590 empty: function() {
3591   if ( !list ) {
3592     list = [];
3593   }
3594   return this;
3595 },
3596

```

```

3597 // Disable .fire and .add
3598 // Abort any current/pending executions
3599 // Clear all callbacks and values
3600 disable: function() {
3601     locked = queue = [];
3602     list = memory = "";
3603     return this;
3604 },
3605 disabled: function() {
3606     return list;
3607 },
3608
3609 // Disable .fire
3610 // Also disable .add unless we have memory (since it would have no effect)
3611 // Abort any pending executions
3612 lock: function() {
3613     locked = queue = [];
3614     if ( !memory && !firing ) {
3615         list = memory = "";
3616     }
3617     return this;
3618 },
3619 locked: function() {
3620     return !locked;
3621 },
3622
3623 // Call all callbacks with the given context and arguments
3624 fireWith: function( context, args ) {
3625     if ( !locked ) {
3626         args = args || [];
3627         args = [ context, args.slice ? args.slice() : args ];
3628         queue.push( args );
3629         if ( !firing ) {
3630             fire();
3631         }
3632     }
3633     return this;
3634 },
3635
3636 // Call all the callbacks with the given arguments
3637 fire: function() {
3638     self.fireWith( this, arguments );
3639     return this;
3640 },
3641
3642 // To know if the callbacks have already been called at least once
3643 fired: function() {
3644     return !!fired;
3645 }
3646 };
3647
3648 return self;
3649 };
3650
3651
3652 function Identity( v ) {
3653     return v;
3654 }
3655
3656 function Thrower( ex ) {
3657     throw ex;
3658 }
3659
3660 function adoptValue( value, resolve, reject, noValue ) {
3661     var method;
3662
3663     try {
3664         // Check for promise aspect first to privilege synchronous behavior
3665         if ( value && isFunction( ( method = value.promise ) ) ) {
3666             method.call( value ).done( resolve ).fail( reject );
3667         }
3668         // Other thenables
3669     } else if ( value && isFunction( ( method = value.then ) ) ) {
3670         method.call( value, resolve, reject );
3671     }
3672     // Other non-thenables
3673     } else {
3674         // Control `resolve` arguments by letting Array#slice cast boolean `noValue` to integer:
3675         // * false: [ value ].slice( 0 ) => resolve( value )
3676         // * true: [ value ].slice( 1 ) => resolve()
3677         resolve.apply( undefined, [ value ].slice( noValue ? 1 : 0 ) );
3678     }
3679 }
3680
3681 // For Promises/A+, convert exceptions into rejections
3682 // Since jQuery.when doesn't unwrap thenables, we can skip the extra checks appearing in
3683 // Deferred#then to conditionally suppress rejection.
3684 } catch ( value ) {
3685     // Support: Android 4.0 only
3686     // Strict mode functions invoked without .call/.apply get global-object context
3687     reject.apply( undefined, [ value ] );
3688 }
3689 }
3690 }
3691
3692 jQuery.extend( {
3693     Deferred: function( func ) {
3694         var tuples = [
3695             // action, add listener, callbacks, [final state]
3696             [ "notify", "progress", jQuery.Callbacks( "memory" ),
3697               jQuery.Callbacks( "memory" ), 2 ],
3698             [ "resolve", "done", jQuery.Callbacks( "once memory" ),
3699               jQuery.Callbacks( "once memory" ), 0, "resolved" ],
3700             [ "reject", "fail", jQuery.Callbacks( "once memory" ),
3701               jQuery.Callbacks( "once memory" ), 1, "rejected" ]
3702         ],
3703         state = "pending",
3704         promise = {
3705             state: function() {
3706                 return state;
3707             },
3708             always: function() {
3709                 return this;
3710             },
3711             always: function() {

```

```

3712 deferred.done( arguments ) fail( arguments );
3713 return this;
3714 },
3715 "catch": function( fn ) {
3716   return promise.then( null, fn );
3717 },
3718
3719 // Keep pipe for back-compat
3720 pipe: function( /* fnDone, fnFail, fnProgress */ ) {
3721   var fns = arguments;
3722
3723   return jQuery.Deferred( function( newDefer ) {
3724     jQuery.each( tuples, function( i, tuple ) {
3725
3726       // Map tuples (progress, done, fail) to arguments (done, fail, progress)
3727       var fn = isFunction( fns[ tuple[ 4 ] ] ) && fns[ tuple[ 4 ] ];
3728
3729       // deferred.progress(function() { bind to newDefer or newDefer.notify })
3730       // deferred.done(function() { bind to newDefer or newDefer.resolve })
3731       // deferred.fail(function() { bind to newDefer or newDefer.reject })
3732       deferred[ tuple[ 1 ] ]( function() {
3733         var returned = fn && fn.apply( this, arguments );
3734         if ( returned && isFunction( returned.promise() ) ) {
3735           returned.promise()
3736             .progress( newDefer.notify )
3737             .done( newDefer.resolve )
3738             .fail( newDefer.reject );
3739         } else {
3740           newDefer[ tuple[ 0 ] + "With" ](
3741             this,
3742             fn ? [ returned ] : arguments
3743           );
3744         }
3745       } );
3746     } );
3747     fns = null;
3748   } ).promise();
3749 },
3750 then: function( onFulfilled, onRejected, onProgress ) {
3751   var maxDepth = 0;
3752   function resolve( depth, deferred, handler, special ) {
3753     return function() {
3754       var that = this;
3755       args = arguments;
3756       mightThrow = function() {
3757         var returned, then;
3758
3759         // Support: Promises/A+ section 2.3.3.3
3760         // https://promisesaplus.com/#point-59
3761         // Ignore double-resolution attempts
3762         if ( depth < maxDepth ) {
3763           return;
3764         }
3765
3766         returned = handler.apply( that, args );
3767
3768         // Support: Promises/A+ section 2.3.1
3769         // https://promisesaplus.com/#point-48
3770         if ( returned === deferred.promise() ) {
3771           throw new TypeError( "Thenable self-resolution" );
3772         }
3773
3774         // Support: Promises/A+ sections 2.3.3.1, 3.5
3775         // https://promisesaplus.com/#point-54
3776         // https://promisesaplus.com/#point-75
3777         // Retrieve `then` only once
3778         then = returned &&
3779
3780           // Support: Promises/A+ section 2.3.4
3781           // https://promisesaplus.com/#point-64
3782           // Only check objects and functions for thenability
3783           ( typeof returned === "object" ||
3784             typeof returned === "function" ) &&
3785             returned.then;
3786
3787         // Handle a returned thenable
3788         if ( isFunction( then ) ) {
3789           // Special processors (notify) just wait for resolution
3790           if ( special ) {
3791             then.call(
3792               returned,
3793               resolve( maxDepth, deferred, Identity, special ),
3794               resolve( maxDepth, deferred, Thrower, special )
3795             );
3796           }
3797           // Normal processors (resolve) also hook into progress
3798         } else {
3799           // ...and disregard older resolution values
3800           maxDepth++;
3801
3802           then.call(
3803             returned,
3804             resolve( maxDepth, deferred, Identity, special ),
3805             resolve( maxDepth, deferred, Thrower, special ),
3806             resolve( maxDepth, deferred, Identity,
3807               deferred.notifyWith )
3808           );
3809         }
3810       };
3811     };
3812
3813     // Handle all other returned values
3814   } else {
3815     // Only substitute handlers pass on context
3816     // and multiple values (non-spec behavior)
3817     if ( handler !== Identity ) {
3818       that = undefined;
3819       args = [ returned ];
3820     }
3821
3822     // Process the value(s)
3823     // Default process is resolve
3824     ( special || deferred.resolveWith ) ( that, args );
3825   }
3826 }

```



```

3827     },
3828
3829     // Only normal processors (resolve) catch and reject exceptions
3830     process = special ?
3831     mightThrow :
3832     function() {
3833         try {
3834             mightThrow();
3835         } catch ( e ) {
3836
3837             if ( jQuery.Deferred.exceptionHook ) {
3838                 jQuery.Deferred.exceptionHook( e,
3839                     process.stackTrace );
3840             }
3841
3842             // Support: Promises/A+ section 2.3.3.3.4.1
3843             // https://promisesaplus.com/#point-61
3844             // Ignore post-resolution exceptions
3845             if ( depth + 1 >= maxDepth ) {
3846
3847                 // Only substitute handlers pass on context
3848                 // and multiple values (non-spec behavior)
3849                 if ( handler !== Thrower ) {
3850                     that = undefined;
3851                     args = [ e ];
3852                 }
3853
3854                 deferred.rejectWith( that, args );
3855             }
3856         };
3857     };
3858
3859     // Support: Promises/A+ section 2.3.3.3.1
3860     // https://promisesaplus.com/#point-57
3861     // Re-resolve promises immediately to dodge false rejection from
3862     // subsequent errors
3863     if ( depth ) {
3864         process();
3865     } else {
3866
3867         // Call an optional hook to record the stack, in case of exception
3868         // since it's otherwise lost when execution goes async
3869         if ( jQuery.Deferred.getStackHook ) {
3870             process.stackTrace = jQuery.Deferred.getStackHook();
3871         }
3872         window.setTimeout( process );
3873     }
3874 };
3875 },
3876
3877 return jQuery.Deferred( function() {
3878
3879     // progress_handlers.add( ... )
3880     tuples[ 0 ][ 3 ].add(
3881         resolve(
3882             0,
3883             newDefer,
3884             isFunction( onProgress ) ?
3885             onProgress :
3886             Identity,
3887             newDefer.notifyWith
3888         )
3889     );
3890
3891     // fulfilled_handlers.add( ... )
3892     tuples[ 1 ][ 3 ].add(
3893         resolve(
3894             0,
3895             newDefer,
3896             isFunction( onFulfilled ) ?
3897             onFulfilled :
3898             Identity
3899         )
3900     );
3901
3902     // rejected_handlers.add( ... )
3903     tuples[ 2 ][ 3 ].add(
3904         resolve(
3905             0,
3906             newDefer,
3907             isFunction( onRejected ) ?
3908             onRejected :
3909             Thrower
3910         )
3911     );
3912 } ).promise();
3913 },
3914
3915 // Get a promise for this deferred
3916 // If obj is provided, the promise aspect is added to the object
3917 promise: function( obj ) {
3918     return obj != null ? jQuery.extend( obj, promise ) : promise;
3919 }
3920 },
3921 deferred = {} );
3922
3923 // Add list-specific methods
3924 jQuery.each( tuples, function( i, tuple ) {
3925     var list = tuple[ 2 ],
3926         stateString = tuple[ 5 ];
3927
3928     // promise.progress = list.add
3929     // promise.done = list.add
3930     // promise.fail = list.add
3931     promise[ tuple[ 1 ] ] = list.add;
3932
3933     // Handle state
3934     if ( stateString ) {
3935         list.add(
3936             function() {
3937
3938                 // state = "resolved" (i.e., fulfilled)
3939                 // state = "rejected"
3940                 state = stateString;
3941

```

```

3942
3943 // rejected_callbacks.disable
3944 // fulfilled_callbacks.disable
3945 tuples[ 3 - i ][ 2 ].disable,
3946
3947 // rejected_handlers.disable
3948 // fulfilled_handlers.disable
3949 tuples[ 3 - i ][ 3 ].disable,
3950
3951 // progress_callbacks.lock
3952 tuples[ 0 ][ 2 ].lock,
3953
3954 // progress_handlers.lock
3955 tuples[ 0 ][ 3 ].lock
3956 };
3957 }
3958
3959 // progress_handlers.fire
3960 // fulfilled_handlers.fire
3961 // rejected_handlers.fire
3962 list.add( tuple[ 3 ].fire );
3963
3964 // deferred.notify = function() { deferred.notifyWith(...) }
3965 // deferred.resolve = function() { deferred.resolveWith(...) }
3966 // deferred.reject = function() { deferred.rejectWith(...) }
3967 deferred[ tuple[ 0 ] ] = function() {
3968     deferred[ tuple[ 0 ] ] + "With" }( this === deferred ? undefined : this, arguments );
3969     return this;
3970 };
3971
3972 // deferred.notifyWith = list.fireWith
3973 // deferred.resolveWith = list.fireWith
3974 // deferred.rejectWith = list.fireWith
3975 deferred[ tuple[ 0 ] ] + "With" ] = list.fireWith;
3976 } );
3977
3978 // Make the deferred a promise
3979 promise.promise( deferred );
3980
3981 // Call given func if any
3982 if ( func ) {
3983     func.call( deferred, deferred );
3984 }
3985
3986 // All done!
3987 return deferred;
3988 },
3989
3990 // Deferred helper
3991 when: function( singleValue ) {
3992     var
3993
3994     // count of uncompleted subordinates
3995     remaining = arguments.length,
3996
3997     // count of unprocessed arguments
3998     i = remaining,
3999
4000     // subordinate fulfillment data
4001     resolveContexts = Array( i ),
4002     resolveValues = slice.call( arguments ),
4003
4004     // the primary Deferred
4005     primary = jQuery.Deferred(),
4006
4007     // subordinate callback factory
4008     updateFunc = function( i ) {
4009         return function( value ) {
4010             resolveContexts[ i ] = this;
4011             resolveValues[ i ] = arguments.length > 1 ? slice.call( arguments ) : value;
4012             if ( !( --remaining ) ) {
4013                 primary.resolveWith( resolveContexts, resolveValues );
4014             }
4015         };
4016     };
4017
4018     // Single- and empty arguments are adopted like Promise.resolve
4019     if ( remaining <= 1 ) {
4020         adoptValue( singleValue, primary.done( updateFunc( i ) ).resolve, primary.reject,
4021             !remaining );
4022
4023         // Use .then() to unwrap secondary thenables (cf. gh-3000)
4024         if ( primary.state() === "pending" ||
4025             isFunction( resolveValues[ i ] ) && resolveValues[ i ].then ) { } {
4026
4027             return primary.then();
4028         }
4029     }
4030
4031     // Multiple arguments are aggregated like Promise.all array elements
4032     while ( i-- ) {
4033         adoptValue( resolveValues[ i ], updateFunc( i ), primary.reject );
4034     }
4035
4036     return primary.promise();
4037 }
4038 } );
4039
4040
4041 // These usually indicate a programmer mistake during development,
4042 // warn about them ASAP rather than swallowing them by default.
4043 var errorNames = /^(Eval|Internal|Range|Reference|Syntax|Type|URI)Error$/;
4044
4045 jQuery.Deferred.exceptionHook = function( error, stack ) {
4046
4047     // Support: IE 8 - 9 only
4048     // Console exists when dev tools are open, which can happen at any time
4049     if ( window.console && window.console.warn && error && errorNames.test( error.name ) ) {
4050         window.console.warn( "jQuery.Deferred exception: " + error.message, error.stack, stack );
4051     }
4052 };
4053
4054
4055
4056

```

```

4057 jQuery.readyException = function( error ) {
4058   window.setTimeout( function() {
4059     throw error;
4060   } );
4061 };
4062
4063
4064
4065
4066 // The deferred used on DOM ready
4067 var readyList = jQuery.Deferred();
4068
4069 jQuery.fn.ready = function( fn ) {
4070   readyList
4071     .then( fn )
4072     .catch( function( error ) {
4073       // Wrap jQuery.readyException in a function so that the lookup
4074       // happens at the time of error handling instead of callback
4075       // registration.
4076       jQuery.readyException( error );
4077     } );
4078   return this;
4079 };
4080
4081 jQuery.extend( {
4082   // Is the DOM ready to be used? Set to true once it occurs.
4083   isReady: false,
4084   // A counter to track how many items to wait for before
4085   // the ready event fires. See #6781
4086   readyWait: 1,
4087   // Handle when the DOM is ready
4088   ready: function( wait ) {
4089     // Abort if there are pending holds or we're already ready
4090     if ( wait === true ? --jQuery.readyWait : jQuery.isReady ) {
4091       return;
4092     }
4093     // Remember that the DOM is ready
4094     jQuery.isReady = true;
4095     // If a normal DOM Ready event fired, decrement, and wait if need be
4096     if ( wait !== true && --jQuery.readyWait > 0 ) {
4097       return;
4098     }
4099     // If there are functions bound, to execute
4100     readyList.resolveWith( document, [ jQuery ] );
4101   };
4102 };
4103
4104 jQuery.ready.then = readyList.then;
4105
4106 // The ready event handler and self cleanup method
4107 function completed() {
4108   document.removeEventListener( "DOMContentLoaded", completed );
4109   window.removeEventListener( "load", completed );
4110   jQuery.ready();
4111 }
4112
4113 // Catch cases where $(document).ready() is called
4114 // after the browser event has already occurred.
4115 // Support: IE <=9 - 10 only
4116 // Older IE sometimes signals "interactive" too soon
4117 if ( document.readyState === "complete" ||
4118   ( document.readyState !== "loading" && document.documentElement.doScroll ) ) {
4119   // Handle it asynchronously to allow scripts the opportunity to delay ready
4120   window.setTimeout( jQuery.ready );
4121 }
4122 else {
4123   // Use the handy event callback
4124   document.addEventListener( "DOMContentLoaded", completed );
4125   // A fallback to window.onload, that will always work
4126   window.addEventListener( "load", completed );
4127 }
4128
4129
4130
4131
4132
4133 // Multifunctional method to get and set values of a collection
4134 // The value/s can optionally be executed if it's a function
4135 var access = function( elems, fn, key, value, chainable, emptyGet, raw ) {
4136   var i = 0,
4137       len = elems.length,
4138       bulk = key == null;
4139
4140   // Sets many values
4141   if ( toType( key ) === "object" ) {
4142     chainable = true;
4143     for ( i in key ) {
4144       access( elems, fn, i, key[i], true, emptyGet, raw );
4145     }
4146   }
4147   // Sets one value
4148   } else if ( value !== undefined ) {
4149     chainable = true;
4150     if ( !isFunction( value ) ) {
4151       raw = true;
4152     }
4153     if ( bulk ) {
4154       // Bulk operations run against the entire set
4155       if ( raw ) {
4156         fn.call( elems, value );

```

```

4172     fn = null;
4173
4174     // ...except when executing function values
4175     } else {
4176         bulk = fn;
4177         fn = function( elem, _key, value ) {
4178             return bulk.call( jQuery( elem ), value );
4179         };
4180     }
4181 }
4182
4183 if ( fn ) {
4184     for ( ; i < len; i++ ) {
4185         fn(
4186             elems[ i ], key, raw ?
4187                 value :
4188                 value.call( elems[ i ], i, fn( elems[ i ], key ) )
4189         );
4190     }
4191 }
4192 }
4193
4194 if ( chainable ) {
4195     return elems;
4196 }
4197
4198 // Gets
4199 if ( bulk ) {
4200     return fn.call( elems );
4201 }
4202
4203 return len ? fn( elems[ 0 ], key ) : emptyGet;
4204 };
4205
4206
4207 // Matches dashed string for camelizing
4208 var rmsPrefix = /^-ms-/;
4209 rdashAlpha = /-([a-z])/g;
4210
4211 // Used by camelCase as callback to replace()
4212 function fcamelCase( _all, letter ) {
4213     return letter.toUpperCase();
4214 }
4215
4216 // Convert dashed to camelCase; used by the css and data modules
4217 // Support: IE <=9 - 11, Edge 12 - 15
4218 // Microsoft forgot to hump their vendor prefix (#9572)
4219 function camelCase( string ) {
4220     return string.replace( rmsPrefix, "ms-" ).replace( rdashAlpha, fcamelCase );
4221 }
4222 var acceptData = function( owner ) {
4223
4224     // Accepts only:
4225     // - Node
4226     // - Node.ELEMENT_NODE
4227     // - Node.DOCUMENT_NODE
4228     // - Object
4229     // - Any
4230     return owner.nodeType === 1 || owner.nodeType === 9 || !( +owner.nodeType );
4231 };
4232
4233
4234
4235
4236 function Data() {
4237     this.expando = jQuery.expando + Data.uid++;
4238 }
4239
4240 Data.uid = 1;
4241
4242 Data.prototype = {
4243
4244     cache: function( owner ) {
4245
4246         // Check if the owner object already has a cache
4247         var value = owner[ this.expando ];
4248
4249         // If not, create one
4250         if ( !value ) {
4251             value = {};
4252
4253             // We can accept data for non-element nodes in modern browsers,
4254             // but we should not, see #8335.
4255             // Always return an empty object.
4256             if ( acceptData( owner ) ) {
4257
4258                 // If it is a node unlikely to be stringify-ed or looped over
4259                 // use plain assignment
4260                 if ( owner.nodeType ) {
4261                     owner[ this.expando ] = value;
4262                 }
4263                 // Otherwise secure it in a non-enumerable property
4264                 // configurable must be true to allow the property to be
4265                 // deleted when data is removed
4266             } else {
4267                 Object.defineProperty( owner, this.expando, {
4268                     value: value,
4269                     configurable: true
4270                 });
4271             }
4272         }
4273     },
4274
4275     return value;
4276 },
4277 set: function( owner, data, value ) {
4278     var prop;
4279     cache = this.cache( owner );
4280
4281     // Handle: [ owner, key, value ] args
4282     // Always use camelCase key (gh-2257)
4283     if ( typeof data === "string" ) {
4284         cache[ camelCase( data ) ] = value;
4285     }
4286     // Handle: [ owner, { properties } ] args

```

```

4287     } else {
4288
4289         // Copy the properties one-by-one to the cache object
4290         for ( prop in data ) {
4291             cache[ camelCase( prop ) ] = data[ prop ];
4292         }
4293     }
4294     return cache;
4295 },
4296 get: function( owner, key ) {
4297     return key === undefined ?
4298         this.cache[ owner ] :
4299
4300         // Always use camelCase key (gh-2257)
4301         owner[ this.expando ] && owner[ this.expando ][ camelCase( key ) ];
4302 },
4303 access: function( owner, key, value ) {
4304
4305     // In cases where either:
4306     //
4307     // 1. No key was specified
4308     // 2. A string key was specified, but no value provided
4309     //
4310     // Take the "read" path and allow the get method to determine
4311     // which value to return, respectively either:
4312     //
4313     // 1. The entire cache object
4314     // 2. The data stored at the key
4315     //
4316     if ( key === undefined ||
4317         ( ! key && typeof key === "string" ) && value === undefined ) {
4318
4319         return this.get( owner, key );
4320     }
4321
4322     // When the key is not a string, or both a key and value
4323     // are specified, set or extend (existing objects) with either:
4324     //
4325     // 1. An object of properties
4326     // 2. A key and value
4327     //
4328     this.set( owner, key, value );
4329
4330     // Since the "set" path can have two possible entry points
4331     // return the expected data based on which path was taken[*]
4332     return value !== undefined ? value : key;
4333 },
4334 remove: function( owner, key ) {
4335     var i,
4336         cache = owner[ this.expando ];
4337
4338     if ( cache === undefined ) {
4339         return;
4340     }
4341
4342     if ( key !== undefined ) {
4343
4344         // Support array or space separated string of keys
4345         if ( Array.isArray( key ) ) {
4346
4347             // If key is an array of keys...
4348             // We always set camelCase keys, so remove that.
4349             key = key.map( camelCase );
4350         } else {
4351             key = camelCase( key );
4352
4353             // If a key with the spaces exists, use it.
4354             // Otherwise, create an array by matching non-whitespace
4355             key = key in cache ?
4356                 [ key ] :
4357                 ( key.match( /\s+/ ) || [] );
4358         }
4359
4360         i = key.length;
4361
4362         while ( i-- ) {
4363             delete cache[ key[ i ] ];
4364         }
4365     }
4366
4367     // Remove the expando if there's no more data
4368     if ( key === undefined || jQuery.isEmptyObject( cache ) ) {
4369
4370         // Support: Chrome <=35 - 45
4371         // Webkit & Blink performance suffers when deleting properties
4372         // from DOM nodes, so set to undefined instead
4373         // https://bugs.chromium.org/p/chromium/issues/detail?id=378607 (bug restricted)
4374         if ( owner.nodeType ) {
4375             owner[ this.expando ] = undefined;
4376         } else {
4377             delete owner[ this.expando ];
4378         }
4379     },
4380     hasData: function( owner ) {
4381         var cache = owner[ this.expando ];
4382         return cache !== undefined && !jQuery.isEmptyObject( cache );
4383     }
4384 };
4385
4386 var dataPriv = new Data();
4387
4388 var dataUser = new Data();
4389
4390
4391
4392 // Implementation Summary
4393 //
4394 // 1. Enforce API surface and semantic compatibility with 1.9.x branch
4395 // 2. Improve the module's maintainability by reducing the storage
4396 //    paths to a single mechanism.
4397 // 3. Use the same single mechanism to support "private" and "user" data.
4398 // 4. _Never_ expose "private" data to user code (TODO: Drop _data, _removeData)
4399 // 5. Avoid exposing implementation details on user objects (eg. expando properties)
4400 // 6. Provide a clear path for implementation upgrade to WeakMap in 2014
4401

```

```

4402 var rbrace = /^(?!\{([wW]*)\}|N([wW]*)\})$/;
4403 rmultiDash = /[A-Z]/g;
4404
4405 function getData( data ) {
4406   if ( data === "true" ) {
4407     return true;
4408   }
4409
4410   if ( data === "false" ) {
4411     return false;
4412   }
4413
4414   if ( data === "null" ) {
4415     return null;
4416   }
4417
4418   // Only convert to a number if it doesn't change the string
4419   if ( data === +data + "" ) {
4420     return +data;
4421   }
4422
4423   if ( rbrace.test( data ) ) {
4424     return JSON.parse( data );
4425   }
4426
4427   return data;
4428 }
4429
4430 function dataAttr( elem, key, data ) {
4431   var name;
4432
4433   // If nothing was found internally, try to fetch any
4434   // data from the HTML5 data-* attribute
4435   if ( data === undefined && elem.nodeType === 1 ) {
4436     name = "data-" + key.replace( rmultiDash, "-$&" ).toLowerCase();
4437     data = elem.getAttribute( name );
4438
4439     if ( typeof data === "string" ) {
4440       try {
4441         data = getData( data );
4442       } catch ( e ) {}
4443
4444       // Make sure we set the data so it isn't changed later
4445       dataUser.set( elem, key, data );
4446     } else {
4447       data = undefined;
4448     }
4449   }
4450   return data;
4451 }
4452
4453 jQuery.extend( {
4454   hasData: function( elem ) {
4455     return dataUser.hasData( elem ) || dataPriv.hasData( elem );
4456   },
4457
4458   data: function( elem, name, data ) {
4459     return dataUser.access( elem, name, data );
4460   },
4461
4462   removeData: function( elem, name ) {
4463     dataUser.remove( elem, name );
4464   },
4465
4466   // TODO: Now that all calls to _data and _removeData have been replaced
4467   // with direct calls to dataPriv methods, these can be deprecated.
4468   _data: function( elem, name, data ) {
4469     return dataPriv.access( elem, name, data );
4470   },
4471
4472   _removeData: function( elem, name ) {
4473     dataPriv.remove( elem, name );
4474   }
4475   });
4476
4477 jQuery.fn.extend( {
4478   data: function( key, value ) {
4479     var i, name, data;
4480     elem = this[ 0 ];
4481     attrs = elem && elem.attributes;
4482
4483     // Gets all values
4484     if ( key === undefined ) {
4485       if ( this.length ) {
4486         data = dataUser.get( elem );
4487
4488         if ( elem.nodeType === 1 && !dataPriv.get( elem, "hasDataAttrs" ) ) {
4489           i = attrs.length;
4490           while ( i-- ) {
4491
4492             // Support: IE 11 only
4493             // The attrs elements can be null (#14894)
4494             if ( attrs[ i ] ) {
4495               name = attrs[ i ].name;
4496               if ( name.indexOf( "data-" ) === 0 ) {
4497                 name = camelCase( name.slice( 5 ) );
4498                 dataAttr( elem, name, data[ name ] );
4499               }
4500             }
4501           }
4502           dataPriv.set( elem, "hasDataAttrs", true );
4503         }
4504       }
4505
4506       return data;
4507     }
4508
4509     // Sets multiple values
4510     if ( typeof key === "object" ) {
4511       return this.each( function() {
4512         dataUser.set( this, key );
4513       } );
4514     }
4515
4516     return access( this, function( value ) {

```

```

4517     var data
4518
4519     // The calling jQuery object (element matches) is not empty
4520     // (and therefore has an element appears at this[ 0 ]) and the
4521     // `value` parameter was not undefined. An empty jQuery object
4522     // will result in `undefined` for elem = this[ 0 ] which will
4523     // throw an exception if an attempt to read a data cache is made.
4524     if ( elem && value === undefined ) {
4525
4526         // Attempt to get data from the cache
4527         // The key will always be camelCased in Data
4528         data = dataUser.get( elem, key );
4529         if ( data !== undefined ) {
4530             return data;
4531         }
4532
4533         // Attempt to "discover" the data in
4534         // HTML5 custom data-* attrs
4535         data = dataAttr( elem, key );
4536         if ( data !== undefined ) {
4537             return data;
4538         }
4539
4540         // We tried really hard, but the data doesn't exist.
4541         return;
4542     }
4543
4544     // Set the data...
4545     this.each( function() {
4546
4547         // We always store the camelCased key
4548         dataUser.set( this, key, value );
4549     } );
4550     }, null, value, arguments.length > 1, null, true );
4551 },
4552
4553 removeData: function( key ) {
4554     return this.each( function() {
4555         dataUser.remove( this, key );
4556     } );
4557 }
4558 );
4559
4560
4561 jQuery.extend( {
4562     queue: function( elem, type, data ) {
4563         var queue;
4564
4565         if ( elem ) {
4566             type = ( type || "fx" ) + "queue";
4567             queue = dataPriv.get( elem, type );
4568
4569             // Speed up dequeue by getting out quickly if this is just a lookup
4570             if ( data ) {
4571                 if ( !queue || Array.isArray( data ) ) {
4572                     queue = dataPriv.access( elem, type, jQuery.makeArray( data ) );
4573                 } else {
4574                     queue.push( data );
4575                 }
4576             }
4577             return queue || [];
4578         }
4579     },
4580
4581     dequeue: function( elem, type ) {
4582         type = type || "fx";
4583
4584         var queue = jQuery.queue( elem, type ),
4585             startLength = queue.length,
4586             fn = queue.shift(),
4587             hooks = jQuery._queueHooks( elem, type ),
4588             next = function() {
4589                 jQuery.dequeue( elem, type );
4590             };
4591
4592         // If the fx queue is dequeued, always remove the progress sentinel
4593         if ( fn === "inprogress" ) {
4594             fn = queue.shift();
4595             startLength--;
4596         }
4597
4598         if ( fn ) {
4599
4600             // Add a progress sentinel to prevent the fx queue from being
4601             // automatically dequeued
4602             if ( type === "fx" ) {
4603                 queue.unshift( "inprogress" );
4604             }
4605
4606             // Clear up the last queue stop function
4607             delete hooks.stop;
4608             fn.call( elem, next, hooks );
4609         }
4610
4611         if ( !startLength && hooks ) {
4612             hooks.empty.fire();
4613         }
4614     },
4615
4616     // Not public - generate a queueHooks object, or return the current one
4617     _queueHooks: function( elem, type ) {
4618         var key = type + "queueHooks";
4619         return dataPriv.get( elem, key ) || dataPriv.access( elem, key, {
4620             empty: jQuery.Callbacks( "once memory" ).add( function() {
4621                 dataPriv.remove( elem, [ type + "queue", key ] );
4622             } )
4623         } );
4624     },
4625     };
4626
4627 jQuery.fn.extend( {
4628     queue: function( type, data ) {
4629         var setter = 2;
4630
4631         if ( typeof type !== "string" ) {

```

```

4632 data = type;
4633 type = "fx";
4634 setter--;
4635 }
4636
4637 if ( arguments.length < setter ) {
4638     return jQuery.queue( this[ 0 ], type );
4639 }
4640
4641 return data === undefined ?
4642     this :
4643     this.each( function() {
4644         var queue = jQuery.queue( this, type, data );
4645
4646         // Ensure a hooks for this queue
4647         jQuery._queueHooks( this, type );
4648
4649         if ( type !== "fx" && queue[ 0 ] !== "inprogress" ) {
4650             jQuery.dequeue( this, type );
4651         }
4652     });
4653 },
4654 dequeue: function( type ) {
4655     return this.each( function() {
4656         jQuery.dequeue( this, type );
4657     });
4658 },
4659 clearQueue: function( type ) {
4660     return this.queue( type || "fx", [] );
4661 },
4662
4663 // Get a promise resolved when queues of a certain type
4664 // are emptied (fx is the type by default)
4665 promise: function( type, obj ) {
4666     var tmp;
4667     count = 1;
4668     defer = jQuery.Deferred();
4669     elements = this;
4670     i = this.length;
4671     resolve = function() {
4672         if ( !count ) {
4673             defer.resolveWith( elements, [ elements ] );
4674         }
4675     };
4676
4677     if ( typeof type !== "string" ) {
4678         obj = type;
4679         type = undefined;
4680     }
4681     type = type || "fx";
4682
4683     while ( i-- ) {
4684         tmp = dataPriv.get( elements[ i ], type + "queueHooks" );
4685         if ( tmp && tmp.empty ) {
4686             count++;
4687             tmp.empty.add( resolve );
4688         }
4689     }
4690     resolve();
4691     return defer.promise( obj );
4692 }
4693 );
4694 var pnum = ( /[+-]?(?:\d*\.|)\d+(?:[eE][+-]?[d+])? ) source;
4695
4696 var rcssNum = new RegExp( "^(?:[+-])=|(" + pnum + ")([a-z%]*)$" , "i" );
4697
4698 var cssExpand = [ "Top", "Right", "Bottom", "Left" ];
4699
4700 var documentElement = document.documentElement;
4701
4702
4703
4704
4705 var isAttached = function( elem ) {
4706     return jQuery.contains( elem.ownerDocument, elem );
4707 };
4708 composed = { composed: true };
4709
4710 // Support: IE 9 - 11+, Edge 12 - 18+, iOS 10.0 - 10.2 only
4711 // Check attachment across shadow DOM boundaries when possible (gh-3504)
4712 // Support: iOS 10.0-10.2 only
4713 // Early iOS 10 versions support `attachShadow` but not `getRootNode`,
4714 // leading to errors. We need to check for `getRootNode`.
4715 if ( documentElement.getRootNode ) {
4716     isAttached = function( elem ) {
4717         return jQuery.contains( elem.ownerDocument, elem ) ||
4718             elem.getRootNode( composed ) === elem.ownerDocument;
4719     };
4720 }
4721 var isHiddenWithinTree = function( elem, el ) {
4722
4723     // isHiddenWithinTree might be called from jQuery#filter function;
4724     // in that case, element will be second argument
4725     elem = el || elem;
4726
4727     // Inline style trumps all
4728     return elem.style.display === "none" ||
4729         elem.style.display === "" &&
4730
4731         // Otherwise, check computed style
4732         // Support: Firefox <= 43 - 45
4733         // Disconnected elements can have computed display: none, so first confirm that elem is
4734         // in the document.
4735         isAttached( elem ) &&
4736
4737         jQuery.css( elem, "display" ) === "none";
4738 };
4739
4740
4741
4742 function adjustCSS( elem, prop, valueParts, tween ) {
4743     var adjusted, scale;
4744     maxIterations = 20;
4745     currentValue = tween ?
4746         function() {

```



```

4747     return tween.cur();
4748   };
4749   function() {
4750     return jQuery.css( elem, prop, "" );
4751   },
4752   initial = currentValue(),
4753   unit = valueParts[ 3 ] || ( jQuery.cssNumber[ prop ] ? "" : "px" ),
4754
4755   // Starting value computation is required for potential unit mismatches
4756   initialInUnit = elem.nodeType &&
4757   ( jQuery.cssNumber[ prop ] || unit !== "px" && +initial ) &&
4758   rcssNum.exec( jQuery.css( elem, prop ) );
4759
4760   if ( initialInUnit && initialInUnit[ 3 ] !== unit ) {
4761
4762     // Support: Firefox <=54
4763     // Halve the iteration target value to prevent interference from CSS upper bounds (gh-2144)
4764     initial = initial / 2;
4765
4766     // Trust units reported by jQuery.css
4767     unit = unit || initialInUnit[ 3 ];
4768
4769     // Iteratively approximate from a nonzero starting point
4770     initialInUnit = +initial || 1;
4771
4772     while ( maxIterations-- ) {
4773
4774       // Evaluate and update our best guess (doubling guesses that zero out).
4775       // Finish if the scale equals or crosses 1 (making the old*new product non-positive).
4776       jQuery.style( elem, prop, initialInUnit + unit );
4777       if ( ( 1 - scale ) * ( 1 - ( scale = currentValue() / initial || 0.5 ) ) <= 0 ) {
4778         maxIterations = 0;
4779       }
4780       initialInUnit = initialInUnit / scale;
4781     }
4782   }
4783
4784   initialInUnit = initialInUnit * 2;
4785   jQuery.style( elem, prop, initialInUnit + unit );
4786
4787   // Make sure we update the tween properties later on
4788   valueParts = valueParts || [];
4789 }
4790
4791 if ( valueParts ) {
4792   initialInUnit = +initialInUnit || +initial || 0;
4793
4794   // Apply relative offset (+/=) if specified
4795   adjusted = valueParts[ 1 ] ?
4796     initialInUnit + ( valueParts[ 1 ] + 1 ) * valueParts[ 2 ] :
4797     +valueParts[ 2 ];
4798   if ( tween ) {
4799     tween.unit = unit;
4800     tween.start = initialInUnit;
4801     tween.end = adjusted;
4802   }
4803 }
4804 return adjusted;
4805 }
4806
4807 var defaultDisplayMap = {};
4808
4809 function getDefaultDisplay( elem ) {
4810   var temp;
4811   doc = elem.ownerDocument;
4812   nodeName = elem.nodeName;
4813   display = defaultDisplayMap[ nodeName ];
4814
4815   if ( display ) {
4816     return display;
4817   }
4818
4819   temp = doc.body.appendChild( doc.createElement( nodeName ) );
4820   display = jQuery.css( temp, "display" );
4821   temp.parentNode.removeChild( temp );
4822
4823   if ( display === "none" ) {
4824     display = "block";
4825   }
4826   defaultDisplayMap[ nodeName ] = display;
4827
4828   return display;
4829 }
4830
4831 function showHide( elements, show ) {
4832   var display, elem,
4833       values = [],
4834       index = 0,
4835       length = elements.length;
4836
4837   // Determine new display value for elements that need to change
4838   for ( ; index < length; index++ ) {
4839     elem = elements[ index ];
4840     if ( !elem.style ) {
4841       continue;
4842     }
4843     display = elem.style.display;
4844     if ( show ) {
4845       // Since we force visibility upon cascade-hidden elements, an immediate (and slow)
4846       // check is required in this first loop unless we have a nonempty display value (either
4847       // inline or about-to-be-restored)
4848       if ( display === "none" ) {
4849         values[ index ] = dataPriv.get( elem, "display" ) || null;
4850         if ( !values[ index ] ) {
4851           elem.style.display = "";
4852         }
4853       }
4854       if ( elem.style.display === "" && isHiddenWithinTree( elem ) ) {
4855         values[ index ] = getDefaultDisplay( elem );
4856       }
4857     } else {
4858       if ( display !== "none" ) {
4859         values[ index ] = display;
4860       }
4861     }
4862   }

```

```

4862     if ( display !== "none" ) {
4863         values[ index ] = "none";
4864     }
4865     // Remember what we're overwriting
4866     dataPriv.set( elem, "display", display );
4867 }
4868 }
4869 }
4870
4871 // Set the display of the elements in a second loop to avoid constant reflow
4872 for ( index = 0; index < length; index++ ) {
4873     if ( values[ index ] !== null ) {
4874         elements[ index ].style.display = values[ index ];
4875     }
4876 }
4877
4878 return elements;
4879 }
4880
4881 jQuery.fn.extend( {
4882     show: function() {
4883         return showHide( this, true );
4884     },
4885     hide: function() {
4886         return showHide( this );
4887     },
4888     toggle: function( state ) {
4889         if ( typeof state === "boolean" ) {
4890             return state ? this.show() : this.hide();
4891         }
4892
4893         return this.each( function() {
4894             if ( isHiddenWithinTree( this ) ) {
4895                 jQuery( this ).show();
4896             } else {
4897                 jQuery( this ).hide();
4898             }
4899         } );
4900     }
4901 };
4902
4903 var rcheckboxableType = ( /^?(?:checkbox|radio)$/i );
4904
4905 var rtagName = ( /<([a-z][^\/0-9\t\nf]*)/i );
4906
4907 var rscriptType = ( /^$|^module$|\/(?:java|ecma)script/i );
4908
4909 ( function() {
4910     var fragment = document.createElement( "div" ),
4911         div = fragment.appendChild( document.createElement( "div" ) ),
4912         input = document.createElement( "input" );
4913
4914     // Support: Android 4.0 - 4.3 only
4915     // Check state lost if the name is set (#11217)
4916     // Support: Windows Web Apps (WWA)
4917     // 'name' and 'type' must use .setAttribute for WWA (#14901)
4918     input.setAttribute( "type", "radio" );
4919     input.setAttribute( "checked", "checked" );
4920     input.setAttribute( "name", "t" );
4921
4922     div.appendChild( input );
4923
4924     // Support: Android <=4.1 only
4925     // Older WebKit doesn't clone checked state correctly in fragments
4926     support.checkClone = div.cloneNode( true ).cloneNode( true ).lastChild.checked;
4927
4928     // Support: IE <=11 only
4929     // Make sure textarea (and checkbox) defaultValue is properly cloned
4930     div.innerHTML = "<textarea>x</textarea>";
4931     support.noCloneChecked = !div.cloneNode( true ).lastChild.defaultValue;
4932
4933     // Support: IE <=9 only
4934     // IE <=9 replaces <option> tags with their contents when inserted outside of
4935     // the select element.
4936     div.innerHTML = "<option></option>";
4937     support.option = !div.lastChild;
4938 } )();
4939
4940 // We have to close these tags to support XHTML (#13200)
4941 var wrapMap = {
4942
4943     // XHTML parsers do not magically insert elements in the
4944     // same way that tag soup parsers do. So we cannot shorten
4945     // this by omitting <tbody> or other required elements.
4946     thead: [ 1, "<table>", "</table>" ],
4947     col: [ 2, "<table><colgroup>", "</colgroup></table>" ],
4948     tr: [ 2, "<table><tbody>", "</tbody></table>" ],
4949     td: [ 3, "<table><tbody><tr>", "</tr></tbody></table>" ],
4950
4951     _default: [ 0, "", "" ]
4952 };
4953
4954 wrapMap.tbody = wrapMap.tfoot = wrapMap.colgroup = wrapMap.caption = wrapMap.thead;
4955 wrapMap.th = wrapMap.td;
4956
4957 // Support: IE <=9 only
4958 if ( !support.option ) {
4959     wrapMap.optgroup = wrapMap.option = [ 1, "<select multiple='multiple'>", "</select>" ];
4960 }
4961
4962 function getAll( context, tag ) {
4963
4964     // Support: IE <=9 - 11 only
4965     // Use typeOf to avoid zero-argument method invocation on host objects (#15151)
4966     var ret;
4967
4968     if ( typeof context.getElementsByTagName !== "undefined" ) {
4969         ret = context.getElementsByTagName( tag || "*" );
4970     } else if ( typeof context.querySelectorAll !== "undefined" ) {
4971         ret = context.querySelectorAll( tag || "*" );
4972     }
4973 }

```

```

4977 } else {
4978   ret = [];
4979 }
4980
4981 if ( tag === undefined || tag && nodeName || context.tag ) {
4982   return jQuery.merge( [ context ], ret );
4983 }
4984
4985 return ret;
4986 }
4987
4988 // Mark scripts as having already been evaluated
4989 function setGlobalEval( elems, refElements ) {
4990   var i = 0,
4991       l = elems.length;
4992   for ( ; i < l; i++ ) {
4993     dataPriv.set(
4994       elems[ i ],
4995       "globalEval",
4996       !refElements || dataPriv.get( refElements[ i ], "globalEval" )
4997     );
4998   }
4999 }
5000
5001 }
5002
5003 var rhtml = /<|&#?w+;/;
5004
5005 function buildFragment( elems, context, scripts, selection, ignored ) {
5006   var elem, tmp, tag, wrap, attached, j,
5007       fragment = context.createDocumentFragment(),
5008       nodes = [],
5009       i = 0,
5010       l = elems.length;
5011   for ( ; i < l; i++ ) {
5012     elem = elems[ i ];
5013     if ( elem || elem === 0 ) {
5014       // Add nodes directly
5015       if ( !jQuery.nodeName( elem, "object" ) ) {
5016         // Support: Android <=4.0 only, PhantomJS 1 only
5017         // push.apply(_, arraylike) throws on ancient WebKit
5018         jQuery.merge( nodes, elem.nodeType ? [ elem ] : elem );
5019       }
5020       // Convert non-html into a text node
5021       } else if ( !rhtml.test( elem ) ) {
5022         nodes.push( context.createTextNode( elem ) );
5023       }
5024       // Convert html into DOM nodes
5025       } else {
5026         tmp = tmp || fragment.appendChild( context.createElement( "div" ) );
5027         // Deserialize a standard representation
5028         tag = ( !jQuery.nodeName( elem, "script" ) ) ? jQuery.nodeName( elem, "script" ) : "div";
5029         wrap = wrapMap[ tag ] || wrapMap.default;
5030         tmp.innerHTML = wrap[ 1 ] + jQuery.htmlPrefilter( elem ) + wrap[ 2 ];
5031         // Descend through wrappers to the right content
5032         j = wrap[ 0 ];
5033         while ( j-- ) {
5034           tmp = tmp.lastChild;
5035         }
5036         // Support: Android <=4.0 only, PhantomJS 1 only
5037         // push.apply(_, arraylike) throws on ancient WebKit
5038         jQuery.merge( nodes, tmp.childNodes );
5039         // Remember the top-level container
5040         tmp = fragment.firstChild;
5041         // Ensure the created nodes are orphaned (#12392)
5042         tmp.textContent = "";
5043       }
5044     }
5045     // Remove wrapper from fragment
5046     fragment.textContent = "";
5047     i = 0;
5048     while ( ( elem = nodes[ i++ ] ) ) {
5049       // Skip elements already in the context collection (trac-4087)
5050       if ( selection && jQuery.inArray( elem, selection ) > -1 ) {
5051         if ( ignored ) {
5052           ignored.push( elem );
5053         }
5054         continue;
5055       }
5056       attached = isAttached( elem );
5057       // Append to fragment
5058       tmp = getAll( fragment.appendChild( elem ), "script" );
5059       // Preserve script evaluation history
5060       if ( attached ) {
5061         setGlobalEval( tmp );
5062       }
5063       // Capture executables
5064       if ( !scripts ) {
5065         j = 0;
5066         while ( ( elem = tmp[ j++ ] ) ) {
5067           if ( !jQuery.nodeName( elem, "script" ) ) {
5068             scripts.push( elem );
5069           }
5070         }
5071       }
5072     }
5073   }
5074 }

```

```

5092     return fragment;
5093 }
5094
5095
5096 var rtypenamespace = /^([\^.]*)?(?:\.(.+))$/;
5097
5098 function returnTrue() {
5099     return true;
5100 }
5101
5102 function returnFalse() {
5103     return false;
5104 }
5105
5106 // Support: IE <=9 - 11+
5107 // focus() and blur() are asynchronous, except when they are no-op.
5108 // So expect focus to be synchronous when the element is already active,
5109 // and blur to be synchronous when the element is not already active.
5110 // (focus and blur are always synchronous in other supported browsers,
5111 // this just defines when we can count on it).
5112 function expectSync( elem, type ) {
5113     return ( elem === safeActiveElement() ) === ( type === "focus" );
5114 }
5115
5116 // Support: IE <=9 only
5117 // Accessing document.activeElement can throw unexpectedly
5118 // https://bugs.jquery.com/ticket/13393
5119 function safeActiveElement() {
5120     try {
5121         return document.activeElement;
5122     } catch ( err ) {}
5123 }
5124
5125 function on( elem, types, selector, data, fn, one ) {
5126     var origFn, type;
5127
5128     // Types can be a map of types/handlers
5129     if ( typeof types === "object" ) {
5130
5131         // ( types-Object, selector, data )
5132         if ( typeof selector !== "string" ) {
5133
5134             // ( types-Object, data )
5135             data = selector || selector;
5136             selector = undefined;
5137         }
5138         for ( type in types ) {
5139             on( elem, type, selector, data, types[ type ], one );
5140         }
5141         return elem;
5142     }
5143
5144     if ( data == null && fn == null ) {
5145
5146         // ( types, fn )
5147         fn = selector;
5148         data = selector = undefined;
5149     } else if ( fn == null ) {
5150         if ( typeof selector === "string" ) {
5151
5152             // ( types, selector, fn )
5153             fn = data;
5154             data = undefined;
5155         } else {
5156
5157             // ( types, data, fn )
5158             fn = data;
5159             data = selector;
5160             selector = undefined;
5161         }
5162     }
5163     if ( fn === false ) {
5164         fn = returnFalse;
5165     } else if ( !fn ) {
5166         return elem;
5167     }
5168
5169     if ( one === 1 ) {
5170         origFn = fn;
5171         fn = function( event ) {
5172
5173             // Can use an empty set, since event contains the info
5174             jQuery().off( event );
5175             return origFn.apply( this, arguments );
5176         };
5177
5178         // Use same guid so caller can remove using origFn
5179         fn.guid = origFn.guid || ( origFn.guid = jQuery.guid++ );
5180     }
5181     return elem.each( function() {
5182         jQuery.event.add( this, types, fn, data, selector );
5183     } );
5184 }
5185
5186 /*
5187  * Helper functions for managing events -- not part of the public interface.
5188  * Props to Dean Edwards' addEvent library for many of the ideas.
5189  */
5190 jQuery.event = {
5191
5192     global: {},
5193
5194     add: function( elem, types, handler, data, selector ) {
5195
5196         var handleObjIn, eventHandle, tmp,
5197             events, t, handleObj,
5198             special, handlers, type, namespaces, origType,
5199             elemData = dataPriv.get( elem );
5200
5201         // Only attach events to objects that accept data
5202         if ( !acceptData( elem ) ) {
5203             return;
5204         }
5205
5206         // Caller can pass in an object of custom data in lieu of the handler

```

```

5207 if ( handler handler ) {
5208     handleObjIn = handler;
5209     handler = handleObjIn handler;
5210     selector = handleObjIn selector;
5211 }
5212
5213 // Ensure that invalid selectors throw exceptions at attach time
5214 // Evaluate against documentElement in case elem is a non-element node (e.g., document)
5215 if ( selector ) {
5216     jQuery.find.matchesSelector( documentElement, selector );
5217 }
5218
5219 // Make sure that the handler has a unique ID, used to find/remove it later
5220 if ( !handler.guid ) {
5221     handler.guid = jQuery.guid++;
5222 }
5223
5224 // Init the element's event structure and main handler, if this is the first
5225 if ( !( events = elemData.events ) ) {
5226     events = elemData.events = Object.create( null );
5227 }
5228 if ( !( eventHandle = elemData.handle ) ) {
5229     eventHandle = elemData.handle = function( e ) {
5230
5231         // Discard the second event of a jQuery.event.trigger() and
5232         // when an event is called after a page has unloaded
5233         return typeof jQuery !== "undefined" && jQuery.event.triggered !== e.type ?
5234             jQuery.event.dispatch.apply( elem, arguments ) : undefined;
5235     };
5236 }
5237
5238 // Handle multiple events separated by a space
5239 types = ( types || "" ).match( /\s*\w+/ ) || [ "" ];
5240 t = types.length;
5241 while ( t-- ) {
5242     tmp = rtypenamespace.exec( types[ t ] ) || [];
5243     type = origType = tmp[ 1 ];
5244     namespaces = ( tmp[ 2 ] || "" ).split( "." ).sort();
5245
5246     // There *must* be a type, no attaching namespace-only handlers
5247     if ( !type ) {
5248         continue;
5249     }
5250
5251     // If event changes its type, use the special event handlers for the changed type
5252     special = jQuery.event.special[ type ] || {};
5253
5254     // If selector defined, determine special event api type, otherwise given type
5255     type = ( selector ? special.delegateType : special.bindType ) || type;
5256
5257     // Update special based on newly reset type
5258     special = jQuery.event.special[ type ] || {};
5259
5260     // handleObj is passed to all event handlers
5261     handleObj = jQuery.extend( {
5262         type: type,
5263         origType: origType,
5264         data: data,
5265         handler: handler,
5266         guid: handler.guid,
5267         selector: selector,
5268         needsContext: selector && jQuery.expr.match.needsContext.test( selector ),
5269         namespace: namespaces.join( "." )
5270     }, handleObjIn );
5271
5272     // Init the event handler queue if we're the first
5273     if ( !handlers = events[ type ] ) {
5274         handlers = events[ type ] = [];
5275         handlers.delegateCount = 0;
5276
5277         // Only use addEventListener if the special events handler returns false
5278         if ( !special.setup || special.setup.call( elem, data, namespaces, eventHandle ) !== false ) {
5279             if ( elem.addEventListener ) {
5280                 elem.addEventListener( type, eventHandle );
5281             }
5282         }
5283     }
5284
5285     if ( special.add ) {
5286         special.add.call( elem, handleObj );
5287
5288         if ( !handleObj.handler.guid ) {
5289             handleObj.handler.guid = handler.guid;
5290         }
5291     }
5292
5293     // Add to the element's handler list, delegates in front
5294     if ( selector ) {
5295         handlers.splice( handlers.delegateCount++, 0, handleObj );
5296     } else {
5297         handlers.push( handleObj );
5298     }
5299
5300     // Keep track of which events have ever been used, for event optimization
5301     jQuery.event.global[ type ] = true;
5302
5303     },
5304
5305     // Detach an event or set of events from an element
5306     remove: function( elem, types, handler, selector, mappedTypes ) {
5307
5308         var j, origCount, tmp;
5309         events = t, handleObj;
5310         special = handlers, type, namespaces, origType;
5311         elemData = dataPriv.hasData( elem ) && dataPriv.get( elem );
5312
5313         if ( !elemData || !( events = elemData.events ) ) {
5314             return;
5315         }
5316
5317         // Once for each type.namespace in types; type may be omitted
5318         types = ( types || "" ).match( /\s*\w+/ ) || [ "" ];

```

```

5322 t = types.length;
5323 while ( t-- ) {
5324     tmp = rtypenamespaces.exec( types[ t ] ) || [];
5325     type = origType = tmp[ 1 ];
5326     namespaces = ( tmp[ 2 ] || "" ).split( "." ).sort();
5327
5328     // Unbind all events (on this namespace, if provided) for the element
5329     if ( type ) {
5330         for ( type in events ) {
5331             jQuery.event.remove( elem, type + types[ t ], handler, selector, true );
5332         }
5333         continue;
5334     }
5335
5336     special = jQuery.event.special[ type ] || {};
5337     type = ( selector ? special.delegateType : special.bindType ) || type;
5338     handlers = events[ type ] || [];
5339     tmp = tmp[ 2 ] || "";
5340     new RegExp( "(^|\\.)" + namespaces.join( "\\(?:.*\\.)" ) + "(\\.|$)" );
5341
5342     // Remove matching events
5343     origCount = j = handlers.length;
5344     while ( j-- ) {
5345         handleObj = handlers[ j ];
5346
5347         if ( ! mappedTypes || origType === handleObj.origType ) &&
5348             ( ! handler || handler.guid === handleObj.guid ) &&
5349             ( ! tmp || tmp.test( handleObj.namespace ) ) &&
5350             ( ! selector || selector === handleObj.selector ||
5351               selector === "**" && handleObj.selector ) ) {
5352             handlers.splice( j, 1 );
5353
5354             if ( handleObj.selector ) {
5355                 handlers.delegateCount--;
5356             }
5357             if ( special.remove ) {
5358                 special.remove.call( elem, handleObj );
5359             }
5360         }
5361     }
5362
5363     // Remove generic event handler if we removed something and no more handlers exist
5364     // (avoids potential for endless recursion during removal of special event handlers)
5365     if ( origCount && ! handlers.length ) {
5366         if ( ! special.teardown ||
5367             special.teardown.call( elem, namespaces, elemData.handle ) === false ) {
5368             jQuery.removeEvent( elem, type, elemData.handle );
5369         }
5370     }
5371
5372     delete events[ type ];
5373 }
5374 }
5375
5376 // Remove data and the expando if it's no longer used
5377 if ( jQuery.isEmptyObject( events ) ) {
5378     dataPriv.remove( elem, "handle events" );
5379 }
5380 },
5381
5382 dispatch: function( nativeEvent ) {
5383
5384     var i, j, ret, matched, handleObj, handlerQueue;
5385     args = new Array( arguments.length );
5386
5387     // Make a writable jQuery.Event from the native event object
5388     event = jQuery.event.fix( nativeEvent );
5389
5390     handlers = (
5391         dataPriv.get( this, "events" ) || Object.create( null )
5392     );
5393     special = jQuery.event.special[ event.type ] || {};
5394
5395     // Use the fix-ed jQuery.Event rather than the (read-only) native event
5396     args[ 0 ] = event;
5397
5398     for ( i = 1; i < arguments.length; i++ ) {
5399         args[ i ] = arguments[ i ];
5400     }
5401
5402     event.delegateTarget = this;
5403
5404     // Call the preDispatch hook for the mapped type, and let it bail if desired
5405     if ( ! special.preDispatch && special.preDispatch.call( this, event ) === false ) {
5406         return;
5407     }
5408
5409     // Determine handlers
5410     handlerQueue = jQuery.event.handlers.call( this, event, handlers );
5411
5412     // Run delegates first; they may want to stop propagation beneath us
5413     i = 0;
5414     while ( ( ! matched = handlerQueue[ i++ ] ) && ! event.isPropagationStopped() ) {
5415         event.currentTarget = matched.elem;
5416
5417         j = 0;
5418         while ( ( ! handleObj = matched.handlers[ j++ ] ) &&
5419             ! event.isImmediatePropagationStopped() ) {
5420
5421             // If the event is namespaced, then each handler is only invoked if it is
5422             // specially universal or its namespaces are a superset of the event's.
5423             if ( ! event.namespace || handleObj.namespace === false ||
5424                 event.namespace.test( handleObj.namespace ) ) {
5425
5426                 event.handleObj = handleObj;
5427                 event.data = handleObj.data;
5428
5429                 ret = ( ( jQuery.event.special[ handleObj.origType ] || {} ).handle ||
5430                     handleObj.handler ).apply( matched.elem, args );
5431
5432                 if ( ret !== undefined ) {
5433                     if ( ( event.result = ret ) === false ) {
5434                         event.preventDefault();
5435                         event.stopPropagation();
5436                     }
5437                 }
5438             }
5439         }
5440     }
5441 }

```

```

5437     }
5438   }
5439 }
5440 }
5441
5442 // Call the postDispatch hook for the mapped type
5443 if ( special.postDispatch ) {
5444   special.postDispatch.call( this, event );
5445 }
5446
5447 return event.result;
5448 },
5449
5450 handlers: function( event, handlers ) {
5451   var i, handleObj, sel, matchedHandlers, matchedSelectors,
5452       handlerQueue = [],
5453       delegateCount = handlers.delegateCount,
5454       cur = event.target;
5455
5456   // Find delegate handlers
5457   if ( delegateCount &&
5458
5459       // Support: IE <=9
5460       // Black-hole SVG <use> instance trees (trac-13180)
5461       cur.nodeType &&
5462
5463       // Support: Firefox <=42
5464       // Suppress spec-violating clicks indicating a non-primary pointer button (trac-3861)
5465       // https://www.w3.org/TR/DOM-Level-3-Events/#event-type-click
5466       // Support: IE 11 only
5467       // ...but not arrow key "clicks" of radio inputs, which can have 'button' -1 (gh-2343)
5468       !(event.type === "click" && event.button >= 1 ) ) {
5469     for ( ; cur !== this; cur = cur.parentNode || this ) {
5470
5471       // Don't check non-elements (#13208)
5472       // Don't process clicks on disabled elements (#6911, #8165, #11382, #11764)
5473       if ( cur.nodeType === 1 && !(event.type === "click" && cur.disabled === true ) ) {
5474         matchedHandlers = [];
5475         matchedSelectors = {};
5476         for ( i = 0; i < delegateCount; i++ ) {
5477           handleObj = handlers[ i ];
5478
5479           // Don't conflict with Object.prototype properties (#13203)
5480           sel = handleObj.selector + " ";
5481
5482           if ( matchedSelectors[ sel ] === undefined ) {
5483             matchedSelectors[ sel ] = handleObj.needsContext ?
5484               jQuery( sel, this ).index( cur ) > -1 :
5485               jQuery.find( sel, this, null, [ cur ] ).length;
5486           }
5487           if ( matchedSelectors[ sel ] ) {
5488             matchedHandlers.push( handleObj );
5489           }
5490         }
5491         if ( matchedHandlers.length ) {
5492           handlerQueue.push( { elem: cur, handlers: matchedHandlers } );
5493         }
5494       }
5495     }
5496   }
5497
5498   // Add the remaining (directly-bound) handlers
5499   cur = this;
5500   if ( delegateCount < handlers.length ) {
5501     handlerQueue.push( { elem: cur, handlers: handlers.slice( delegateCount ) } );
5502   }
5503
5504   return handlerQueue;
5505 },
5506
5507
5508 addProp: function( name, hook ) {
5509   Object.defineProperty( jQuery.Event.prototype, name, {
5510     enumerable: true,
5511     configurable: true,
5512
5513     get: isFunction( hook ) ?
5514       function() {
5515         if ( this.originalEvent ) {
5516           return hook( this.originalEvent );
5517         }
5518       } :
5519       function() {
5520         if ( this.originalEvent ) {
5521           return this.originalEvent[ name ];
5522         }
5523       },
5524
5525     set: function( value ) {
5526       Object.defineProperty( this, name, {
5527         enumerable: true,
5528         configurable: true,
5529         writable: true,
5530         value: value
5531       });
5532     }
5533   });
5534 },
5535
5536 fix: function( originalEvent ) {
5537   return originalEvent || jQuery.expando ?
5538     originalEvent :
5539     new jQuery.Event( originalEvent );
5540 },
5541
5542 special: {
5543   load: {
5544
5545     // Prevent triggered image.load events from bubbling to window.load
5546     noBubble: true
5547   },
5548   click: {
5549
5550     // Utilize native event to ensure correct state for checkable inputs
5551     setup: function( data ) {

```

```

5552
5553 // For mutual compressibility with _default, replace `this` access with a local var.
5554 // `|| data` is dead code meant only to preserve the variable through minification.
5555 var el = this || data;
5556
5557 // Claim the first handler
5558 if ( rcheckableType test: el type ) &&
5559     el click && nodeName( el, "input" ) ) {
5560
5561     // dataPriv.set( el, "click", ... )
5562     leverageNative( el, "click", returnTrue );
5563 }
5564
5565 // Return false to allow normal processing in the caller
5566 return false;
5567 },
5568 trigger: function( data ) {
5569
5570     // For mutual compressibility with _default, replace `this` access with a local var.
5571     // `|| data` is dead code meant only to preserve the variable through minification.
5572     var el = this || data;
5573
5574     // Force setup before triggering a click
5575     if ( rcheckableType test: el type ) &&
5576         el click && nodeName( el, "input" ) ) {
5577
5578         leverageNative( el, "click" );
5579     }
5580
5581     // Return non-false to allow normal event-path propagation
5582     return true;
5583 },
5584
5585 // For cross-browser consistency, suppress native .click() on links
5586 // Also prevent it if we're currently inside a leveraged native-event stack
5587 _default: function( event ) {
5588     var target = event.target;
5589     return rcheckableType test: target type ) &&
5590         target click && nodeName( target, "input" ) &&
5591         dataPriv.get( target, "click" ) ||
5592         nodeName( target, "a" );
5593 }
5594 },
5595
5596 beforeunload: {
5597     postDispatch: function( event ) {
5598
5599         // Support: Firefox 20+
5600         // Firefox doesn't alert if the returnValue field is not set.
5601         if ( event result !== undefined && event originalEvent ) {
5602             event originalEvent.returnValue = event result;
5603         }
5604     }
5605 },
5606 };
5607
5608 // Ensure the presence of an event listener that handles manually-triggered
5609 // synthetic events by interrupting progress until reinvoked in response to
5610 // *native* events that it fires directly, ensuring that state changes have
5611 // already occurred before other listeners are invoked.
5612 function leverageNative( el, type, expectSync ) {
5613
5614     // Missing expectSync indicates a trigger call, which must force setup through jQuery.event.add
5615     if ( !expectSync ) {
5616         if ( dataPriv.get( el, type ) === undefined ) {
5617             jQuery.event.add( el, type, returnTrue );
5618         }
5619         return;
5620     }
5621
5622     // Register the controller as a special universal handler for all event namespaces
5623     dataPriv.set( el, type, false );
5624     jQuery.event.add( el, type, {
5625         namespace: false,
5626         handler: function( event ) {
5627             var notAsync, result;
5628             saved = dataPriv.get( this, type );
5629
5630             if ( ( event isTrigger & 1 ) && this[ type ] ) {
5631
5632                 // Interrupt processing of the outer synthetic .trigger()ed event
5633                 // Saved data should be false in such cases, but might be a leftover capture object
5634                 // from an async native handler (gh-4350)
5635                 if ( !saved.length ) {
5636
5637                     // Store arguments for use when handling the inner native event
5638                     // There will always be at least one argument (an event object), so this array
5639                     // will not be confused with a leftover capture object.
5640                     saved = slice.call( arguments );
5641                     dataPriv.set( this, type, saved );
5642
5643                     // Trigger the native event and capture its result
5644                     // Support: IE <=9 - 11+
5645                     // focus() and blur() are asynchronous
5646                     notAsync = expectSync( this, type );
5647                     this[ type ]();
5648                     result = dataPriv.get( this, type );
5649                     if ( saved !== result || notAsync ) {
5650                         dataPriv.set( this, type, false );
5651                     } else {
5652                         result = {};
5653                     }
5654                 }
5655                 if ( saved !== result ) {
5656
5657                     // Cancel the outer synthetic event
5658                     event stopImmediatePropagation();
5659                     event preventDefault();
5660
5661                     // Support: Chrome 86+
5662                     // In Chrome, if an element having a focusout handler is blurred by
5663                     // clicking outside of it, it invokes the handler synchronously. If
5664                     // that handler calls `.remove()` on the element, the data is cleared,
5665                     // leaving `result` undefined. We need to guard against this.
5666                     return result && result value;

```



```

5667     }
5668
5669     // If this is an inner synthetic event for an event with a bubbling surrogate
5670     // (focus or blur), assume that the surrogate already propagated from triggering the
5671     // native event and prevent that from happening again here.
5672     // This technically gets the ordering wrong w.r.t. to '.trigger()' (in which the
5673     // bubbling surrogate propagates *after* the non-bubbling base), but that seems
5674     // less bad than duplication.
5675     } else if ( ( jQuery.event.special[ type ] || {} ).delegateType ) {
5676         event.stopPropagation();
5677     }
5678
5679     // If this is a native event triggered above, everything is now in order
5680     // Fire an inner synthetic event with the original arguments
5681     } else if ( saved.length ) {
5682
5683         // ...and capture the result
5684         dataPriv.set( this, type, {
5685             value: jQuery.event.trigger(
5686
5687                 // Support: IE <=9 - 11+
5688                 // Extend with the prototype to reset the above stopImmediatePropagation()
5689                 jQuery.extend( saved[ 0 ], jQuery.Event.prototype ),
5690                 saved.slice( 1 ),
5691                 this
5692             )
5693         } );
5694
5695         // Abort handling of the native event
5696         event.stopImmediatePropagation();
5697     }
5698 }
5699 } );
5700 }
5701
5702 jQuery.removeEvent = function( elem, type, handle ) {
5703
5704     // This "if" is needed for plain objects
5705     if ( elem.removeEventListener ) {
5706         elem.removeEventListener( type, handle );
5707     }
5708 };
5709
5710 jQuery.Event = function( src, props ) {
5711
5712     // Allow instantiation without the 'new' keyword
5713     if ( !( this instanceof jQuery.Event ) ) {
5714         return new jQuery.Event( src, props );
5715     }
5716
5717     // Event object
5718     if ( src && src.type ) {
5719         this.originalEvent = src;
5720         this.type = src.type;
5721
5722         // Events bubbling up the document may have been marked as prevented
5723         // by a handler lower down the tree; reflect the correct value.
5724         this.isDefaultPrevented = src.defaultPrevented ||
5725             src.defaultPrevented === undefined &&
5726
5727             // Support: Android <=2.3 only
5728             src.returnValue === false ?
5729             returnTrue :
5730             returnFalse;
5731
5732         // Create target properties
5733         // Support: Safari <=6 - 7 only
5734         // Target should not be a text node (#504, #13143)
5735         this.target = ( src.target && src.target.nodeType !== 3 ) ?
5736             src.target.parentNode :
5737             src.target;
5738
5739         this.currentTarget = src.currentTarget;
5740         this.relatedTarget = src.relatedTarget;
5741
5742         // Event type
5743     } else {
5744         this.type = src;
5745     }
5746
5747     // Put explicitly provided properties onto the event object
5748     if ( props ) {
5749         jQuery.extend( this, props );
5750     }
5751
5752     // Create a timestamp if incoming event doesn't have one
5753     this.timeStamp = src && src.timeStamp || Date.now();
5754
5755     // Mark it as fixed
5756     this[ jQuery.expando ] = true;
5757 };
5758
5759 // jQuery.Event is based on DOM3 Events as specified by the ECMAScript Language Binding
5760 // https://www.w3.org/TR/2003/WD-DOM-Level-3-Events-20030331/ecma-script-binding.html
5761 jQuery.Event.prototype = {
5762     constructor: jQuery.Event,
5763     isDefaultPrevented: returnFalse,
5764     isPropagationStopped: returnFalse,
5765     isImmediatePropagationStopped: returnFalse,
5766     isSimulated: false,
5767
5768     preventDefault: function() {
5769         var e = this.originalEvent;
5770
5771         this.isDefaultPrevented = returnTrue;
5772
5773         if ( e && this.isSimulated ) {
5774             e.preventDefault();
5775         }
5776     },
5777     stopPropagation: function() {
5778         var e = this.originalEvent;
5779
5780         this.isPropagationStopped = returnTrue;
5781     }

```

```

5782 if ( e && !this.isSimulated ) {
5783   e.stopPropagation();
5784 }
5785 };
5786 stopImmediatePropagation: function() {
5787   var e = this.originalEvent;
5788
5789   this.isImmediatePropagationStopped = returnTrue;
5790
5791   if ( e && !this.isSimulated ) {
5792     e.stopImmediatePropagation();
5793   }
5794
5795   this.stopPropagation();
5796 }
5797 };
5798
5799 // Includes all common event props including KeyEvent and MouseEvent specific props
5800 jQuery.each( {
5801   altKey: true,
5802   bubbles: true,
5803   cancelable: true,
5804   changedTouches: true,
5805   ctrlKey: true,
5806   detail: true,
5807   eventPhase: true,
5808   metaKey: true,
5809   pageX: true,
5810   pageY: true,
5811   shiftKey: true,
5812   view: true,
5813   "char": true,
5814   code: true,
5815   charCode: true,
5816   key: true,
5817   keyCode: true,
5818   button: true,
5819   buttons: true,
5820   clientX: true,
5821   clientY: true,
5822   offsetX: true,
5823   offsetY: true,
5824   pointerId: true,
5825   pointerType: true,
5826   screenX: true,
5827   screenY: true,
5828   targetTouches: true,
5829   toElement: true,
5830   touches: true,
5831   which: true
5832 }, jQuery.event.addProp );
5833
5834 jQuery.each( { focus: "focusin", blur: "focusout" }, function( type, delegateType ) {
5835   jQuery.event.special[ type ] = {
5836
5837     // Utilize native event if possible so blur/focus sequence is correct
5838     setup: function() {
5839
5840       // Claim the first handler
5841       // dataPriv.set( this, "focus", ... )
5842       // dataPriv.set( this, "blur", ... )
5843       leverageNative: this.type === "focusin";
5844
5845       // Return false to allow normal processing in the caller
5846       return false;
5847     },
5848     trigger: function() {
5849
5850       // Force setup before trigger
5851       leverageNative: this.type;
5852
5853       // Return non-false to allow normal event-path propagation
5854       return true;
5855     },
5856
5857     // Suppress native focus or blur as it's already being fired
5858     // in leverageNative.
5859     _default: function() {
5860       return true;
5861     },
5862     delegateType: delegateType
5863   };
5864 });
5865 });
5866
5867 // Create mouseenter/leave events using mouseover/out and event-time checks
5868 // so that event delegation works in jQuery.
5869 // Do the same for pointerenter/pointerleave and pointerover/pointerout
5870 //
5871 // Support: Safari 7 only
5872 // Safari sends mouseenter too often; see:
5873 // https://bugs.chromium.org/p/chromium/issues/detail?id=470258
5874 // for the description of the bug (it existed in older Chrome versions as well).
5875 jQuery.each( {
5876   mouseenter: "mouseover",
5877   mouseleave: "mouseout",
5878   pointerenter: "pointerover",
5879   pointerleave: "pointerout"
5880 }, function( orig, fix ) {
5881   jQuery.event.special[ orig ] = {
5882     delegateType: fix,
5883     bindType: fix,
5884
5885     handle: function( event ) {
5886       var ret,
5887           target = this,
5888           related = event.relatedTarget,
5889           handleObj = event.handleObj;
5890
5891       // For mouseenter/leave call the handler if related is outside the target.
5892       // NB: No relatedTarget if the mouse left/entered the browser window
5893       if ( !related || ( related !== target && jQuery.contains( target, related ) ) ) {
5894         event.type = handleObj.origType;
5895         ret = handleObj.handler.apply( this, arguments );
5896         event.type = fix;

```

```

5897     }
5898     return ret;
5899 }
5900 };
5901 });
5902
5903 jQuery.fn.extend( {
5904
5905     on: function( types, selector, data, fn ) {
5906         return on( this, types, selector, data, fn );
5907     },
5908     one: function( types, selector, data, fn ) {
5909         return on( this, types, selector, data, fn, 1 );
5910     },
5911     off: function( types, selector, fn ) {
5912         var handleObj, type;
5913         if ( types && types.preventDefault && types.handleObj ) {
5914
5915             // ( event ) dispatched jQuery.Event
5916             handleObj = types.handleObj;
5917             jQuery( types.delegateTarget ).off(
5918                 handleObj.namespace ?
5919                     handleObj.origType + "." + handleObj.namespace :
5920                     handleObj.origType,
5921                     handleObj.selector,
5922                     handleObj.handler
5923             );
5924             return this;
5925         }
5926         if ( typeof types === "object" ) {
5927
5928             // ( types-object [, selector] )
5929             for ( type in types ) {
5930                 this.off( type, selector, types[ type ] );
5931             }
5932             return this;
5933         }
5934         if ( selector === false || typeof selector === "function" ) {
5935
5936             // ( types [, fn] )
5937             fn = selector;
5938             selector = undefined;
5939         }
5940         if ( fn === false ) {
5941             fn = returnFalse;
5942         }
5943         return this.each( function() {
5944             jQuery.event.remove( this, types, fn, selector );
5945         } );
5946     }
5947 });
5948
5949 var
5950 // Support: IE <= 10 - 11, Edge 12 - 13 only
5951 // In IE/Edge using regex groups here causes severe slowdowns.
5952 // See https://connect.microsoft.com/IE/feedback/details/1736512/
5953 rnoInnerhtml = /<script|<style|<link/i,
5954 // checked="checked" or checked
5955 rchecked = /checked\s*(?:[^=]|=\s*.checked.)/i,
5956 rcleanScript = /<\s*<!(?:\[CDATA\[|--\])(?:\|)\|-->\s*$/g;
5957
5958 // Prefer a tbody over its parent table for containing new rows
5959 function manipulationTarget( elem, content ) {
5960     if ( nodeName( elem, "table" ) &&
5961         nodeName( content.nodeType !== 11 ? content : content.firstChild, "tr" ) ) {
5962
5963         return jQuery( elem ).children( "tbody" )[ 0 ] || elem;
5964     }
5965
5966     return elem;
5967 }
5968
5969 // Replace/restore the type attribute of script elements for safe DOM manipulation
5970 function disableScript( elem ) {
5971     elem.type = ( elem.getAttribute( "type" ) !== null ) + "/" + elem.type;
5972     return elem;
5973 }
5974 function restoreScript( elem ) {
5975     if ( ( elem.type || "" ).slice( 0, 5 ) === "true/" ) {
5976         elem.type = elem.type.slice( 5 );
5977     } else {
5978         elem.removeAttribute( "type" );
5979     }
5980
5981     return elem;
5982 }
5983
5984 // Copy private data: events, handlers, etc.
5985 function dataPriv( src ) {
5986     if ( dataPriv.hasData( src ) ) {
5987         pdataOld = dataPriv.get( src );
5988         events = pdataOld.events;
5989
5990         if ( events ) {
5991             jQuery.each( events, function( type, handlers ) {
5992                 jQuery.event.remove( dest, type, handlers[ 0 ] );
5993             } );
5994         }
5995
5996         // 2. Copy user data
5997         if ( dataUser.hasData( src ) ) {
5998             uid = dataUser.get( src );
5999             events[ type ] = uid;
6000         }
6001     }
6002 }
6003
6004 // 1. Copy private data: events, handlers, etc.
6005 function dataUser( src ) {
6006     if ( dataUser.hasData( src ) ) {
6007         uid = dataUser.get( src );
6008         events[ type ] = uid;
6009     }
6010 }
6011
6012 // 2. Copy user data
6013 if ( dataUser.hasData( src ) ) {
6014     uid = dataUser.get( src );
6015     events[ type ] = uid;
6016 }

```

```

6012 udataOld = dataUser access( src );
6013 udataCur = jQuery.extend( {}, udataOld );
6014
6015 dataUser.set( dest, udataCur );
6016 }
6017 }
6018
6019 // Fix IE bugs, see support tests
6020 function fixInput( src, dest ) {
6021     var nodeName = dest.nodeName.toLowerCase();
6022
6023     // Fails to persist the checked state of a cloned checkbox or radio button.
6024     if ( nodeName === "input" && rcheckableType.test( src.type ) ) {
6025         dest.checked = src.checked;
6026
6027         // Fails to return the selected option to the default selected state when cloning options
6028     } else if ( nodeName === "input" || nodeName === "textarea" ) {
6029         dest.defaultValue = src.defaultValue;
6030     }
6031 }
6032
6033 function domManip( collection, args, callback, ignored ) {
6034
6035     // Flatten any nested arrays
6036     args = flat( args );
6037
6038     var fragment, first, scripts, hasScripts, node, doc,
6039         i = 0,
6040         l = collection.length,
6041         iNoClone = l - 1,
6042         value = args[ 0 ],
6043         valuesFunction = isFunction( value );
6044
6045     // We can't cloneNode fragments that contain checked, in WebKit
6046     if ( valuesFunction ||
6047         ( l > 1 && typeof value === "string" &&
6048             !support.checkClone && rchecked.test( value ) ) ) {
6049         return collection.each( function( index ) {
6050             var self = collection.eq( index );
6051             if ( valuesFunction ) {
6052                 args[ 0 ] = value.call( this, index, self.html() );
6053             }
6054             domManip( self, args, callback, ignored );
6055         } );
6056     }
6057
6058     if ( l ) {
6059         fragment = buildFragment( args, collection[ 0 ].ownerDocument, false, collection, ignored );
6060         first = fragment.firstChild;
6061
6062         if ( fragment.childNodes.length === 1 ) {
6063             fragment = first;
6064         }
6065
6066         // Require either new content or an interest in ignored elements to invoke the callback
6067         if ( first || ignored ) {
6068             scripts = jQuery.map( getAll( fragment, "script" ), disableScript );
6069             hasScripts = scripts.length;
6070
6071             // Use the original fragment for the last item
6072             // instead of the first because it can end up
6073             // being emptied incorrectly in certain situations (#8070).
6074             for ( ; i < l; i++ ) {
6075                 node = fragment;
6076
6077                 if ( i !== iNoClone ) {
6078                     node = jQuery.clone( node, true, true );
6079
6080                     // Keep references to cloned scripts for later restoration
6081                     if ( hasScripts ) {
6082
6083                         // Support: Android <=4.0 only, PhantomJS 1 only
6084                         // push.apply(_, arraylike) throws on ancient WebKit
6085                         jQuery.merge( scripts, getAll( node, "script" ) );
6086                     }
6087                 }
6088
6089                 callback.call( collection[ i ], node, i );
6090             }
6091
6092             if ( hasScripts ) {
6093                 doc = scripts[ scripts.length - 1 ].ownerDocument;
6094
6095                 // Reenable scripts
6096                 jQuery.map( scripts, restoreScript );
6097
6098                 // Evaluate executable scripts on first document insertion
6099                 for ( i = 0; i < hasScripts; i++ ) {
6100                     node = scripts[ i ];
6101                     if ( !rscriptType.test( node.type ) || !node.src &&
6102                         !dataPriv.access( node, "globalEval" ) &&
6103                         !jQuery.contains( doc, node ) ) {
6104
6105                         if ( !node.src && ( node.type || "" ).toLowerCase() !== "module" ) {
6106
6107                             // Optional AJAX dependency, but won't run scripts if not present
6108                             if ( jQuery._evalUrl && !node.noModule ) {
6109                                 jQuery._evalUrl( node.src, {
6110                                     nonce: node.nonce || node.getAttribute( "nonce" )
6111                                 }, doc );
6112                             }
6113                         } else {
6114                             DOMEval( node.textContent.replace( rcleanScript, "" ), node, doc );
6115                         }
6116                     }
6117                 }
6118             }
6119         }
6120     }
6121
6122     return collection;
6123 }
6124
6125 function remove( elem, selector, keepData ) {
6126     var node;

```

```

6127 nodes = selector ? jQuery filter( selector, elem ) : elem,
6128 i = 0;
6129
6130 for ( ; ( node = nodes[ i ] ) != null; i++ ) {
6131   if ( !keepData && node.nodeType === 1 ) {
6132     jQuery cleanData( getAll( node ) );
6133   }
6134
6135   if ( !node.parentNode ) {
6136     if ( !keepData && isAttached( node ) ) {
6137       setGlobalEval( getAll( node, "script" ) );
6138     }
6139     node.parentNode.removeChild( node );
6140   }
6141 }
6142
6143 return elem;
6144 }
6145
6146 jQuery.extend( {
6147   htmlPrefilter: function( html ) {
6148     return html;
6149   },
6150
6151   clone: function( elem, dataAndEvents, deepDataAndEvents ) {
6152     var i, l, srcElements, destElements,
6153         clone = elem.cloneNode( true ),
6154         inPage = isAttached( elem );
6155
6156     // Fix IE cloning issues
6157     if ( !support.noCloneChecked && ( elem.nodeType === 1 || elem.nodeType === 11 ) &&
6158         !jQuery.isXMLDoc( elem ) ) {
6159
6160       // We eschew Sizzle here for performance reasons: https://jsperf.com/getall-vs-sizzle/2
6161       destElements = getAll( clone );
6162       srcElements = getAll( elem );
6163
6164       for ( i = 0, l = srcElements.length; i < l; i++ ) {
6165         fixInput( srcElements[ i ], destElements[ i ] );
6166       }
6167     }
6168
6169     // Copy the events from the original to the clone
6170     if ( dataAndEvents ) {
6171       if ( deepDataAndEvents ) {
6172         srcElements = srcElements || getAll( elem );
6173         destElements = destElements || getAll( clone );
6174
6175         for ( i = 0, l = srcElements.length; i < l; i++ ) {
6176           cloneCopyEvent( srcElements[ i ], destElements[ i ] );
6177         }
6178       } else {
6179         cloneCopyEvent( elem, clone );
6180       }
6181     }
6182
6183     // Preserve script evaluation history
6184     destElements = getAll( clone, "script" );
6185     if ( destElements.length > 0 ) {
6186       setGlobalEval( destElements, !inPage && getAll( elem, "script" ) );
6187     }
6188
6189     // Return the cloned set
6190     return clone;
6191   },
6192
6193   cleanData: function( elems ) {
6194     var data, elem, type,
6195         special = jQuery.event.special,
6196         i = 0;
6197
6198     for ( ; ( elem = elems[ i ] ) != undefined; i++ ) {
6199       if ( acceptData( elem ) ) {
6200         if ( ( data = elem[ dataPriv.expando ] ) ) {
6201           if ( data.events ) {
6202             for ( type in data.events ) {
6203               if ( special[ type ] ) {
6204                 jQuery.event.remove( elem, type );
6205               }
6206             }
6207           } else {
6208             jQuery.removeEvent( elem, type, data.handle );
6209           }
6210         }
6211       }
6212
6213       // Support: Chrome <=35 - 45+
6214       // Assign undefined instead of using delete, see Data#remove
6215       elem[ dataPriv.expando ] = undefined;
6216     }
6217     if ( elem[ dataUser.expando ] ) {
6218
6219       // Support: Chrome <=35 - 45+
6220       // Assign undefined instead of using delete, see Data#remove
6221       elem[ dataUser.expando ] = undefined;
6222     }
6223   },
6224 },
6225 );
6226
6227
6228 jQuery.fn.extend( {
6229   detach: function( selector ) {
6230     return remove( this, selector, true );
6231   },
6232
6233   remove: function( selector ) {
6234     return remove( this, selector );
6235   },
6236
6237   text: function( value ) {
6238     return access( this, function( value ) {
6239       return value == undefined ?
6240         jQuery.text( this ) :
6241         this.empty().each( function() {

```

```

6242     if ( this.nodeType === 1 || this.nodeType === 11 || this.nodeType === 9 ) {
6243         this.textContent = value;
6244     }
6245     };
6246     }, null, value, arguments.length );
6247     },
6248     append: function() {
6249         return domManip( this, arguments, function( elem ) {
6250             if ( this.nodeType === 1 || this.nodeType === 11 || this.nodeType === 9 ) {
6251                 var target = manipulationTarget( this, elem );
6252                 target.appendChild( elem );
6253             }
6254         } );
6255     },
6256     prepend: function() {
6257         return domManip( this, arguments, function( elem ) {
6258             if ( this.nodeType === 1 || this.nodeType === 11 || this.nodeType === 9 ) {
6259                 var target = manipulationTarget( this, elem );
6260                 target.insertBefore( elem, target.firstChild );
6261             }
6262         } );
6263     },
6264     before: function() {
6265         return domManip( this, arguments, function( elem ) {
6266             if ( this.parentNode ) {
6267                 this.parentNode.insertBefore( elem, this );
6268             }
6269         } );
6270     },
6271     after: function() {
6272         return domManip( this, arguments, function( elem ) {
6273             if ( this.parentNode ) {
6274                 this.parentNode.insertBefore( elem, this.nextSibling );
6275             }
6276         } );
6277     },
6278     empty: function() {
6279         var elem;
6280         for ( ; ( elem = this[ i ] ) != null; i++ ) {
6281             if ( elem.nodeType === 1 ) {
6282                 // Prevent memory leaks
6283                 jQuery.cleanData( getAll( elem, false ) );
6284                 // Remove any remaining nodes
6285                 elem.textContent = "";
6286             }
6287         }
6288         return this;
6289     },
6290     clone: function( dataAndEvents, deepDataAndEvents ) {
6291         dataAndEvents = dataAndEvents == null ? false : dataAndEvents;
6292         deepDataAndEvents = deepDataAndEvents == null ? dataAndEvents : deepDataAndEvents;
6293         return this.map( function() {
6294             return jQuery.clone( this, dataAndEvents, deepDataAndEvents );
6295         } );
6296     },
6297     html: function( value ) {
6298         return access( this, function( value ) {
6299             var elem = this[ 0 ] || {};
6300             i = 0;
6301             l = this.length;
6302             if ( value === undefined && elem.nodeType === 1 ) {
6303                 return elem.innerHTML;
6304             }
6305             // See if we can take a shortcut and just use innerHTML
6306             if ( typeof value === "string" && !rnoinnerHTML.test( value ) &&
6307                 !wrapMap[ ( rtagName.exec( value ) || [ "", "" ] )[ 1 ].toLowerCase() ] ) {
6308                 value = jQuery.htmlPrefilter( value );
6309                 try {
6310                     for ( ; i < l; i++ ) {
6311                         elem = this[ i ] || {};
6312                         // Remove element nodes and prevent memory leaks
6313                         if ( elem.nodeType === 1 ) {
6314                             jQuery.cleanData( getAll( elem, false ) );
6315                             elem.innerHTML = value;
6316                         }
6317                     }
6318                     elem = 0;
6319                 } catch ( e ) {}
6320             }
6321             // If using innerHTML throws an exception, use the fallback method
6322             if ( !elem ) {
6323                 if ( elem ) {
6324                     this.empty().append( value );
6325                 }
6326                 }, null, value, arguments.length );
6327             },
6328             replaceWith: function() {
6329                 var ignored = [];
6330                 // Make the changes, replacing each non-ignored context element with the new content
6331                 return domManip( this, arguments, function( elem ) {
6332                     var parent = this.parentNode;
6333                     if ( jQuery.inArray( this, ignored ) < 0 ) {

```

```

6357     jQuery.cleanData( getAll( this ) );
6358     if ( parent ) {
6359         parent.replaceChild( elem, this );
6360     }
6361 }
6362
6363 // Force callback invocation
6364 }, ignored );
6365 }
6366 });
6367
6368 jQuery.each( {
6369     appendTo: "append",
6370     prependTo: "prepend",
6371     insertBefore: "before",
6372     insertAfter: "after",
6373     replaceAll: "replaceWith"
6374 }, function( name, original ) {
6375     jQuery.fn[ name ] = function( selector ) {
6376         var elems,
6377             ret = [],
6378             insert = jQuery( selector ),
6379             last = insert.length - 1,
6380             i = 0;
6381
6382         for ( ; i <= last; i++ ) {
6383             elems = i === last ? this : this.clone( true );
6384             jQuery( insert[ i ] ).original( i, elems );
6385
6386             // Support: Android <=4.0 only, PhantomJS 1 only
6387             // .get() because push.apply(_, arraylike) throws on ancient WebKit
6388             push.apply( ret, elems.get() );
6389         }
6390
6391         return this.pushStack( ret );
6392     };
6393 });
6394
6395 var rnumnonpx = new RegExp( "^(" + pnum + ")(?!px)[a-z%]+$", "i" );
6396
6397 var getStyles = function( elem ) {
6398     // Support: IE <=11 only, Firefox <=30 (#15098, #14150)
6399     // IE throws on elements created in popups
6400     // FF meanwhile throws on frame elements through "defaultView.getComputedStyle"
6401     var view = elem.ownerDocument.defaultView;
6402
6403     if ( !view || !view.opener ) {
6404         view = window;
6405     }
6406
6407     return view.getComputedStyle( elem );
6408 };
6409
6410 var swap = function( elem, options, callback ) {
6411     var ret, name;
6412     old = {};
6413
6414     // Remember the old values, and insert the new ones
6415     for ( name in options ) {
6416         old[ name ] = elem.style[ name ];
6417         elem.style[ name ] = options[ name ];
6418     }
6419
6420     ret = callback.call( elem );
6421
6422     // Revert the old values
6423     for ( name in options ) {
6424         elem.style[ name ] = old[ name ];
6425     }
6426
6427     return ret;
6428 };
6429
6430 var rboxStyle = new RegExp( cssExpand.join( "|" ), "i" );
6431
6432
6433
6434
6435 ( function() {
6436     // Executing both pixelPosition & boxSizingReliable tests require only one layout
6437     // so they're executed at the same time to save the second computation.
6438     function computeStyleTests() {
6439         // This is a singleton, we need to execute it only once
6440         if ( !div ) {
6441             return;
6442         }
6443
6444         container.style.cssText = "position:absolute;left:-1111px;width:60px;" +
6445             "margin-top:1px;padding:0;border:0";
6446         div.style.cssText =
6447             "position:relative;display:block;box-sizing:border-box;overflow:scroll;" +
6448             "margin:auto;border:1px;padding:1px;" +
6449             "width:60%;top:1%";
6450         documentElement.appendChild( container ).appendChild( div );
6451
6452         var divStyle = window.getComputedStyle( div );
6453         pixelPositionVal = divStyle.top !== "1%";
6454
6455         // Support: Android 4.0 - 4.3 only, Firefox <=3 - 44
6456         reliableMarginLeftVal = roundPixelMeasures( divStyle.marginLeft ) === 12;
6457
6458         // Support: Android 4.0 - 4.3 only, Safari <=9.1 - 10.1, iOS <=7.0 - 9.3
6459         // Some styles come back with percentage values, even though they shouldn't
6460         div.style.right = "60%";
6461         pixelBoxStylesVal = roundPixelMeasures( divStyle.right ) === 36;
6462
6463         // Support: IE 9 - 11 only
6464         // Detect misreporting of content dimensions for box-sizing:border-box elements
6465         boxSizingReliableVal = roundPixelMeasures( divStyle.width ) === 36;
6466
6467         // Support: IE 9 only
6468         // Detect overflow:scroll screwiness (gh-3699)
6469
6470         // Support: Chrome <=64

```

```

6472 // Don't get tricked when zoom affects offsetWidth (gh-4029)
6473 div.style.position = "absolute";
6474 scrollbarSizeVal = roundPixelMeasures( div.offsetWidth / 3 ) === 12;
6475
6476 documentElement.removeChild( container );
6477
6478 // Nullify the div so it wouldn't be stored in the memory and
6479 // it will also be a sign that checks already performed
6480 div = null;
6481 }
6482
6483 function roundPixelMeasures( measure ) {
6484     return Math.round( parseFloat( measure ) );
6485 }
6486
6487 var pixelPositionVal, boxSizingReliableVal, scrollbarSizeVal, pixelBoxStylesVal,
6488     reliableTrDimensionsVal, reliableMarginLeftVal;
6489 container = document.createElement( "div" );
6490 div = document.createElement( "div" );
6491
6492 // Finish early in limited (non-browser) environments
6493 if ( !div.style ) {
6494     return;
6495 }
6496
6497 // Support: IE <= 9 - 11 only
6498 // Style of cloned element affects source element cloned (#8908)
6499 div.style.backgroundClip = "content-box";
6500 div.cloneNode( true ).style.backgroundClip = "";
6501 support.clearCloneStyle = div.style.backgroundClip === "content-box";
6502
6503 jQuery.extend( support, {
6504     boxSizingReliable: function() {
6505         computeStyleTests();
6506         return boxSizingReliableVal;
6507     },
6508     pixelBoxStyles: function() {
6509         computeStyleTests();
6510         return pixelBoxStylesVal;
6511     },
6512     pixelPosition: function() {
6513         computeStyleTests();
6514         return pixelPositionVal;
6515     },
6516     reliableMarginLeft: function() {
6517         computeStyleTests();
6518         return reliableMarginLeftVal;
6519     },
6520     scrollbarSize: function() {
6521         computeStyleTests();
6522         return scrollbarSizeVal;
6523     },
6524
6525     // Support: IE 9 - 11+, Edge 15 - 18+
6526     // IE/Edge misreport `getComputedStyle` of table rows with width/height
6527     // set in CSS while `offset*` properties report correct values.
6528     // Behavior in IE 9 is more subtle than in newer versions & it passes
6529     // some versions of this test; make sure not to make it pass there!
6530     //
6531     // Support: Firefox 70+
6532     // Only Firefox includes border widths
6533     // in computed dimensions. (gh-4529)
6534     reliableTrDimensions: function() {
6535         var table, tr, trChild, trStyle;
6536         if ( !reliableTrDimensionsVal ) {
6537             table = document.createElement( "table" );
6538             tr = document.createElement( "tr" );
6539             trChild = document.createElement( "div" );
6540
6541             table.style.cssText = "position:absolute;left:-1111px;border-collapse:separate";
6542             tr.style.cssText = "border:1px solid";
6543
6544             // Support: Chrome 86+
6545             // Height set through cssText does not get applied.
6546             // Computed height then comes back as 0.
6547             tr.style.height = "1px";
6548             trChild.style.height = "9px";
6549
6550             // Support: Android 8 Chrome 86+
6551             // In our bodyBackground.html iframe,
6552             // display for all div elements is set to "inline",
6553             // which causes a problem only in Android 8 Chrome 86.
6554             // Ensuring the div is display: block
6555             // gets around this issue.
6556             trChild.style.display = "block";
6557
6558             documentElement.appendChild( table );
6559             table.appendChild( tr );
6560             tr.appendChild( trChild );
6561
6562             trStyle = window.getComputedStyle( tr );
6563             reliableTrDimensionsVal = ( parseInt( trStyle.height, 10 ) +
6564                 parseInt( trStyle.borderTopWidth, 10 ) +
6565                 parseInt( trStyle.borderBottomWidth, 10 ) ) === tr.offsetHeight;
6566
6567             documentElement.removeChild( table );
6568         }
6569         return reliableTrDimensionsVal;
6570     },
6571     //
6572     //
6573     //
6574     //
6575     function curCSS( elem, name, computed ) {
6576         var width, minWidth, maxWidth, ret;
6577
6578         // Support: Firefox 51+
6579         // Retrieving style before computed somehow
6580         // fixes an issue with getting wrong values
6581         // on detached elements
6582         style = elem.style;
6583
6584         computed = computed || getStyles( elem );
6585
6586

```



```

6587 // getPropertyValue is needed for:
6588 // .css('filter') (IE 9 only, #12537)
6589 // .css('--customProperty) (#3144)
6590 if ( computed ) {
6591   ret = computed.getPropertyValue( name ) || computed[ name ];
6592 }
6593 if ( ret === "" && !isAttached( elem ) ) {
6594   ret = jQuery.style( elem, name );
6595 }
6596 // A tribute to the "awesome hack by Dean Edwards"
6597 // Android Browser returns percentage for some values,
6598 // but width seems to be reliably pixels.
6599 // This is against the CSSOM draft spec:
6600 // https://drafts.csswg.org/cssom/#resolved-values
6601 // https://drafts.csswg.org/cssom/#resolved-values
6602 if ( !support.pixelBoxStyles() && rnumnonpx.test( ret ) && rboxStyle.test( name ) ) {
6603   // Remember the original values
6604   width = style.width;
6605   minWidth = style.minWidth;
6606   maxWidth = style.maxWidth;
6607   // Put in the new values to get a computed value out
6608   style.minWidth = style.maxWidth = style.width = ret;
6609   ret = computed.width;
6610   // Revert the changed values
6611   style.width = width;
6612   style.minWidth = minWidth;
6613   style.maxWidth = maxWidth;
6614 }
6615 }
6616 return ret !== undefined ?
6617 // Support: IE <= 9 - 11 only
6618 // IE returns zIndex value as an integer.
6619 ret + "" :
6620 ret;
6621 }
6622 function addGetHookIf( conditionFn, hookFn ) {
6623   // Define the hook, we'll check on the first run if it's really needed.
6624   return {
6625     get: function() {
6626       if ( conditionFn() ) {
6627         // Hook not needed (or it's not possible to use it due
6628         // to missing dependency), remove it.
6629         delete this.get;
6630         return;
6631       }
6632       // Hook needed; redefine it so that the support test is not executed again.
6633       return ( this.get = hookFn ).apply( this, arguments );
6634     };
6635   };
6636 }
6637 var cssPrefixes = [ "Webkit", "Moz", "ms" ],
6638     emptyStyle = document.createElement( "div" ).style,
6639     vendorProps = {};
6640 // Return a vendor-prefixed property or undefined
6641 function vendorPropName( name ) {
6642   // Check for vendor prefixed names
6643   var capName = name[ 0 ].toUpperCase() + name.slice( 1 ),
6644       i = cssPrefixes.length;
6645   while ( i-- ) {
6646     name = cssPrefixes[ i ] + capName;
6647     if ( name in emptyStyle ) {
6648       return name;
6649     }
6650   }
6651 }
6652 // Return a potentially-mapped jQuery.cssProps or vendor prefixed property
6653 function finalPropName( name ) {
6654   var final = jQuery.cssProps[ name ] || vendorProps[ name ];
6655   if ( final ) {
6656     return final;
6657   }
6658   if ( name in emptyStyle ) {
6659     return name;
6660   }
6661   return vendorProps[ name ] = vendorPropName( name ) || name;
6662 }
6663 var
6664 // Swappable if display is none or starts with table
6665 // except "table", "table-cell", or "table-caption"
6666 // See here for display values: https://developer.mozilla.org/en-US/docs/CSS/display
6667 rdisplayswap = /^(none|table(?!-c[ea]).+)/,
6668 rcustomProp = /^--/,
6669 cssShow = { position: "absolute", visibility: "hidden", display: "block" },
6670 cssNormalTransform = {
6671   letterSpacing: "0",
6672   fontWeight: "400"
6673 };
6674 function setPositiveNumber( _elem, value, subtract ) {
6675   // Any relative (+/-) values have already been
6676   // normalized at this point
6677   var matches = rcssNum.exec( value );
6678   return matches ?

```

```

6702 // Guard against undefined "subtract", e.g., when used as in cssHooks
6703 Math.max( 0, matches[ 2 ] - ( subtract[ 0 ] ) + ( matches[ 3 ] || "px" ) :
6704 value
6705 }
6706
6707 function boxModelAdjustment( elem, dimension, box, isBorderBox, styles, computedVal ) {
6708     var i = dimension === "width" ? 1 : 0,
6709         extra = 0,
6710         delta = 0;
6711
6712     // Adjustment may not be necessary
6713     if ( box === ( isBorderBox ? "border" : "content" ) ) {
6714         return 0;
6715     }
6716
6717     for ( ; i < 4; i += 2 ) {
6718
6719         // Both box models exclude margin
6720         if ( box === "margin" ) {
6721             delta += jQuery.css( elem, box + cssExpand[ i ], true, styles );
6722         }
6723
6724         // If we get here with a content-box, we're seeking "padding" or "border" or "margin"
6725         if ( !isBorderBox ) {
6726
6727             // Add padding
6728             delta += jQuery.css( elem, "padding" + cssExpand[ i ], true, styles );
6729
6730             // For "border" or "margin", add border
6731             if ( box !== "padding" ) {
6732                 delta += jQuery.css( elem, "border" + cssExpand[ i ] + "Width", true, styles );
6733             }
6734
6735             // But still keep track of it otherwise
6736             } else {
6737                 extra += jQuery.css( elem, "border" + cssExpand[ i ] + "Width", true, styles );
6738             }
6739
6740             // If we get here with a border-box (content + padding + border), we're seeking "content" or
6741             // "padding" or "margin"
6742             } else {
6743
6744                 // For "content", subtract padding
6745                 if ( box === "content" ) {
6746                     delta -= jQuery.css( elem, "padding" + cssExpand[ i ], true, styles );
6747                 }
6748
6749                 // For "content" or "padding", subtract border
6750                 if ( box !== "margin" ) {
6751                     delta -= jQuery.css( elem, "border" + cssExpand[ i ] + "Width", true, styles );
6752                 }
6753             }
6754
6755             // Account for positive content-box scroll gutter when requested by providing computedVal
6756             if ( !isBorderBox && computedVal >= 0 ) {
6757
6758                 // offsetWidth/offsetHeight is a rounded sum of content, padding, scroll gutter, and border
6759                 // Assuming integer scroll gutter, subtract the rest and round down
6760                 delta += Math.max( 0, Math.ceil(
6761                     elem[ "offset" + dimension[ 0 ].toUpperCase() + "Dimension" ] -
6762                     computedVal -
6763                     delta -
6764                     extra -
6765                     0.5
6766                 ) );
6767
6768                 // If offsetWidth/offsetHeight is unknown, then we can't determine content-box scroll gutter
6769                 // Use an explicit zero to avoid NaN (gh-3964)
6770                 } ) || 0;
6771             }
6772
6773             return delta;
6774         }
6775
6776         function getWidthOrHeight( elem, dimension, extra ) {
6777
6778             // Start with computed style
6779             var styles = getStyles( elem );
6780
6781             // To avoid forcing a reflow, only fetch boxSizing if we need it (gh-4322).
6782             // Fake content-box until we know it's needed to know the true value.
6783             boxSizingNeeded = !support.boxSizingReliable() || extra;
6784             isBorderBox = boxSizingNeeded &&
6785                 jQuery.css( elem, "boxSizing", false, styles ) === "border-box",
6786             valuesBorderBox = isBorderBox;
6787
6788             val = curCSS( elem, dimension, styles );
6789             offsetProp = "offset" + dimension[ 0 ].toUpperCase() + "Dimension" + ( dimension.slice( 1 ) );
6790
6791             // Support: Firefox <=54
6792             // Return a confounding non-pixel value or feign ignorance, as appropriate.
6793             if ( !rnumnonpx.test( val ) ) {
6794                 if ( !extra ) {
6795                     return val;
6796                 }
6797                 val = "auto";
6798             }
6799
6800             // Support: IE 9 - 11 only
6801             // Use offsetWidth/offsetHeight for when box sizing is unreliable.
6802             // In those cases, the computed value can be trusted to be border-box.
6803             if ( ( !support.boxSizingReliable() && isBorderBox ) ||
6804
6805                 // Support: IE 10 - 11+, Edge 15 - 18+
6806                 // IE/Edge misreport `getComputedStyle` of table rows with width/height
6807                 // set in CSS while `offset*` properties report correct values.
6808                 // Interestingly, in some cases IE 9 doesn't suffer from this issue.
6809                 !support.reliableTrDimensions() && nodeName( elem, "tr" ) ) {
6810
6811                 // Fall back to offsetWidth/offsetHeight when value is "auto"
6812                 // This happens for inline elements with no explicit setting (gh-3571)
6813                 val = "auto" ||
6814
6815                 // Support: Android <=4.1 - 4.3 only
6816                 // Also use offsetWidth/offsetHeight for misreported inline dimensions (gh-3602)

```

```

6817 |parseFloat( val ) && jQuery.css( elem, "display", false, styles ) === "inline" ) &&
6818
6819 // Make sure the element is visible & connected
6820 elem.getBoundingClientRect().length ) {
6821
6822 isBorderBox = jQuery.css( elem, "boxSizing", false, styles ) === "border-box";
6823
6824 // Where available, offsetWidth/offsetHeight approximate border box dimensions.
6825 // Where not available (e.g., SVG), assume unreliable box-sizing and interpret the
6826 // retrieved value as a content box dimension.
6827 valuesBorderBox = offsetProp in elem;
6828 if ( valuesBorderBox ) {
6829     val = elem[ offsetProp ];
6830 }
6831 }
6832
6833 // Normalize "" and auto
6834 val = parseFloat( val ) || 0;
6835
6836 // Adjust for the element's box model
6837 return ( val +
6838     boxModelAdjustment(
6839         elem,
6840         dimension,
6841         extra || ( isBorderBox ? "border" : "content" ),
6842         valuesBorderBox,
6843         styles
6844
6845         // Provide the current computed size to request scroll gutter calculation (gh-3589)
6846         val
6847     ) + "px";
6848 }
6849 }
6850
6851 jQuery.extend( {
6852
6853     // Add in style property hooks for overriding the default
6854     // behavior of getting and setting a style property
6855     cssHooks: {
6856         opacity: {
6857             get: function( elem, computed ) {
6858                 if ( computed ) {
6859
6860                     // We should always get a number back from opacity
6861                     var ret = curCSS( elem, "opacity" );
6862                     return ret === "" ? "1" : ret;
6863                 }
6864             }
6865         },
6866     },
6867
6868     // Don't automatically add "px" to these possibly-unitless properties
6869     cssNumber: {
6870         "animationIterationCount": true,
6871         "columnCount": true,
6872         "fillOpacity": true,
6873         "flexGrow": true,
6874         "flexShrink": true,
6875         "fontWeight": true,
6876         "gridArea": true,
6877         "gridColumn": true,
6878         "gridColumnEnd": true,
6879         "gridColumnStart": true,
6880         "gridRow": true,
6881         "gridRowEnd": true,
6882         "gridRowStart": true,
6883         "lineHeight": true,
6884         "opacity": true,
6885         "order": true,
6886         "orphans": true,
6887         "widows": true,
6888         "zIndex": true,
6889         "zoom": true
6890     },
6891
6892     // Add in properties whose names you wish to fix before
6893     // setting or getting the value
6894     cssProps: {},
6895
6896     // Get and set the style property on a DOM Node
6897     style: function( elem, name, value, extra ) {
6898
6899         // Don't set styles on text and comment nodes
6900         if ( !elem || elem.nodeType === 3 || elem.nodeType === 8 || !elem.style ) {
6901             return;
6902         }
6903
6904         // Make sure that we're working with the right name
6905         var ret, type, hooks,
6906             origName = camelCase( name ),
6907             isCustomProp = rcustomProp.test( name ),
6908             style = elem.style;
6909
6910         // Make sure that we're working with the right name. We don't
6911         // want to query the value if it is a CSS custom property
6912         // since they are user-defined.
6913         if ( !isCustomProp ) {
6914             name = finalPropName( origName );
6915         }
6916
6917         // Gets hook for the prefixed version, then unprefixed version
6918         hooks = jQuery.cssHooks[ name ] || jQuery.cssHooks[ origName ];
6919
6920         // Check if we're setting a value
6921         if ( value !== undefined ) {
6922             type = typeof value;
6923
6924             // Convert "+=" or "-=" to relative numbers (#7345)
6925             if ( type === "string" && ( ret = rcssNum.exec( value ) ) && ret[ 1 ] ) {
6926                 value = adjustCSS( elem, name, ret );
6927
6928                 // Fixes bug #9237
6929                 type = "number";
6930             }
6931

```

```

6932 // Make sure that null and NaN values aren't set (#7116)
6933 if ( value == null || value !== value ) {
6934     return;
6935 }
6936
6937 // If a number was passed in, add the unit (except for certain CSS properties)
6938 // The isCustomProp check can be removed in jQuery 4.0 when we only auto-append
6939 // "px" to a few hardcoded values.
6940 if ( type === "number" && !isCustomProp ) {
6941     value += ret && ret[ 3 ] || ( jQuery.cssNumber[ origName ] ? "" : "px" );
6942 }
6943
6944 // background-* props affect original clone's values
6945 if ( !support.clearCloneStyle && value === "" && name.indexOf( "background" ) === 0 ) {
6946     style[ name ] = "inherit";
6947 }
6948
6949 // If a hook was provided, use that value, otherwise just set the specified value
6950 if ( !hooks || !( "set" in hooks ) ) {
6951     ( value = hooks.set( elem, value, extra ) ) !== undefined ? {
6952
6953         if ( isCustomProp ) {
6954             style.setProperty( name, value );
6955         } else {
6956             style[ name ] = value;
6957         }
6958     } :
6959 } else {
6960     // If a hook was provided get the non-computed value from there
6961     if ( hooks && "get" in hooks &&
6962         ( ret = hooks.get( elem, false, extra ) ) !== undefined ) {
6963
6964         return ret;
6965     }
6966
6967     // Otherwise just get the value from the style object
6968     return style[ name ];
6969 }
6970
6971 },
6972
6973 css: function( elem, name, extra, styles ) {
6974     var val, num, hooks,
6975         origName = camelCase( name ),
6976         isCustomProp = rcustomProp.test( name );
6977
6978     // Make sure that we're working with the right name. We don't
6979     // want to modify the value if it is a CSS custom property
6980     // since they are user-defined.
6981     if ( !isCustomProp ) {
6982         name = finalPropName( origName );
6983     }
6984
6985     // Try prefixed name followed by the unprefixed name
6986     hooks = jQuery.cssHooks[ name ] || jQuery.cssHooks[ origName ];
6987
6988     // If a hook was provided get the computed value from there
6989     if ( hooks && "get" in hooks ) {
6990         val = hooks.get( elem, true, extra );
6991     }
6992
6993     // Otherwise, if a way to get the computed value exists, use that
6994     if ( val === undefined ) {
6995         val = curCSS( elem, name, styles );
6996     }
6997
6998     // Convert "normal" to computed value
6999     if ( val === "normal" && name in cssNormalTransform ) {
7000         val = cssNormalTransform[ name ];
7001     }
7002
7003     // Make numeric if forced or a qualifier was provided and val looks numeric
7004     if ( extra === "" || extra ) {
7005         num = parseFloat( val );
7006         return extra === true || isFinite( num ) ? num || 0 : val;
7007     }
7008
7009     return val;
7010 }
7011
7012 );
7013
7014 jQuery.each( [ "height", "width", function( _i, dimension ) {
7015     jQuery.cssHooks[ dimension ] = {
7016         get: function( elem, computed, extra ) {
7017             if ( computed ) {
7018
7019                 // Certain elements can have dimension info if we invisibly show them
7020                 // but it must have a current display style that would benefit
7021                 return rdisplayswap.test( jQuery.css( elem, "display" ) ) &&
7022
7023                     // Support: Safari 8+
7024                     // Table columns in Safari have non-zero offsetWidth & zero
7025                     // getBoundingClientRect().width unless display is changed.
7026                     // Support: IE <= 11 only
7027                     // Running getBoundingClientRect on a disconnected node
7028                     // in IE throws an error.
7029                     ( !elem.getBoundingClientRect().length || !elem.getBoundingClientRect().width ) ?
7030                     swap( elem, cssShow, function() {
7031                         return getWidthOrHeight( elem, dimension, extra );
7032                     } ) :
7033                     getWidthOrHeight( elem, dimension, extra );
7034             }
7035         },
7036
7037         set: function( elem, value, extra ) {
7038             var matches;
7039             styles = getStyles( elem );
7040
7041             // Only read styles.position if the test has a chance to fail
7042             // to avoid forcing a reflow.
7043             scrollboxSizeBuggy = !support.scrollboxSize() &&
7044                 styles.position === "absolute";
7045
7046             // To avoid forcing a reflow, only fetch boxSizing if we need it (gh-3991)

```

```

7047     boxSizingNeeded = scrollbarSizeBuggy || extra,
7048     isBorderBox = boxSizingNeeded &&
7049     jQuery.css( elem, "boxSizing", false, styles ) === "border-box",
7050     subtract = extra ?
7051     boxModelAdjustment(
7052         elem,
7053         dimension,
7054         extra,
7055         isBorderBox,
7056         styles
7057     ) :
7058     0;
7059
7060     // Account for unreliable border-box dimensions by comparing offset* to computed and
7061     // faking a content-box to get border and padding (gh-3699)
7062     if ( isBorderBox && scrollbarSizeBuggy ) {
7063         subtract -= Math.ceil(
7064             elem[ "offset" + dimension[ 0 ].toUpperCase() + dimension.slice( 1 ) ] -
7065             parseFloat( styles[ dimension ] ) -
7066             boxModelAdjustment( elem, dimension, "border", false, styles ) -
7067             0.5
7068         );
7069     }
7070
7071     // Convert to pixels if value adjustment is needed
7072     if ( subtract && ( matches = rcssNum.exec( value ) ) &&
7073         ( matches[ 3 ] || "px" ) !== "px" ) {
7074
7075         elem.style[ dimension ] = value;
7076         value = jQuery.css( elem, dimension );
7077     }
7078
7079     return setPositiveNumber( elem, value, subtract );
7080 }
7081 };
7082 });
7083
7084 jQuery.cssHooks.marginLeft = addGetHookIf( support.reliableMarginLeft,
7085 function( elem, computed ) {
7086     if ( computed ) {
7087         return ( parseFloat( curCSS( elem, "marginLeft" ) ) ||
7088             elem.getBoundingClientRect().left -
7089             swap( elem, { marginLeft: 0 }, function() {
7090                 return elem.getBoundingClientRect().left;
7091             } )
7092         ) + "px";
7093     }
7094 }
7095 );
7096
7097 // These hooks are used by animate to expand properties
7098 jQuery.each( [
7099     "margin",
7100     "padding",
7101     "border", "Width"
7102 ], function( prefix, suffix ) {
7103     jQuery.cssHooks[ prefix + suffix ] = {
7104         expand: function( value ) {
7105             var i = 0,
7106                 expanded = {};
7107
7108             // Assumes a single number if not a string
7109             parts = typeof value === "string" ? value.split( " " ) : [ value ];
7110
7111             for ( ; i < 4; i++ ) {
7112                 expanded[ prefix + cssExpand[ i ] + suffix ] =
7113                     parts[ i ] || parts[ i - 2 ] || parts[ 0 ];
7114             }
7115
7116             return expanded;
7117         }
7118     };
7119 });
7120
7121 if ( prefix !== "margin" ) {
7122     jQuery.cssHooks[ prefix + suffix ].set = setPositiveNumber;
7123 }
7124
7125 jQuery.fn.extend( {
7126     css: function( name, value ) {
7127         return access( this, function( elem, name, value ) {
7128             var styles, len,
7129                 map = {},
7130                 i = 0;
7131
7132             if ( Array.isArray( name ) ) {
7133                 styles = getStyles( elem );
7134                 len = name.length;
7135
7136                 for ( ; i < len; i++ ) {
7137                     map[ name[ i ] ] = jQuery.css( elem, name[ i ], false, styles );
7138                 }
7139
7140                 return map;
7141             }
7142
7143             return value !== undefined ?
7144                 jQuery.style( elem, name, value ) :
7145                 jQuery.css( elem, name );
7146         }, name, value, arguments.length > 1 );
7147     }
7148 });
7149
7150 function Tween( elem, options, prop, end, easing ) {
7151     return new Tween.prototype.init( elem, options, prop, end, easing );
7152 }
7153
7154 jQuery.Tween = Tween;
7155
7156 Tween.prototype = {
7157     constructor: Tween,
7158     init: function( elem, options, prop, end, easing, unit ) {
7159         this.elem = elem;
7160         this.prop = prop;
7161         this.easing = easing || jQuery.easing._default;

```

```

7162 this.options = options;
7163 this.start = this.now = this.cur();
7164 this.end = end;
7165 this.unit = unit || ( jQuery.cssNumber[ prop ] ? "" : "px" );
7166 },
7167 cur: function() {
7168     var hooks = Tween.propHooks[ this.prop ];
7169
7170     return hooks && hooks.get ?
7171         hooks.get( this ) :
7172         Tween.propHooks._default.get( this );
7173 },
7174 run: function( percent ) {
7175     var eased;
7176     hooks = Tween.propHooks[ this.prop ];
7177
7178     if ( this.options.duration ) {
7179         this.pos = eased = jQuery.easing[ this.easing ](
7180             percent, this.options.duration * percent, 0, 1, this.options.duration
7181         );
7182     } else {
7183         this.pos = eased = percent;
7184     }
7185     this.now = ( this.end - this.start ) * eased + this.start;
7186
7187     if ( this.options.step ) {
7188         this.options.step.call( this.elem, this.now, this );
7189     }
7190
7191     if ( hooks && hooks.set ) {
7192         hooks.set( this );
7193     } else {
7194         Tween.propHooks._default.set( this );
7195     }
7196     return this;
7197 }
7198 };
7199
7200 Tween.prototype.init.prototype = Tween.prototype;
7201
7202 Tween.propHooks = {
7203     _default: {
7204         get: function( tween ) {
7205             var result;
7206
7207             // Use a property on the element directly when it is not a DOM element,
7208             // or when there is no matching style property that exists.
7209             if ( tween.elem.nodeType !== 1 ||
7210                 tween.elem[ tween.prop ] != null && tween.elem.style[ tween.prop ] == null ) {
7211                 return tween.elem[ tween.prop ];
7212             }
7213
7214             // Passing an empty string as a 3rd parameter to .css will automatically
7215             // attempt a parseFloat and fallback to a string if the parse fails.
7216             // Simple values such as "10px" are parsed to Float;
7217             // complex values such as "rotate(1rad)" are returned as-is.
7218             result = jQuery.css( tween.elem, tween.prop, "" );
7219
7220             // Empty strings, null, undefined and "auto" are converted to 0.
7221             return !result || result === "auto" ? 0 : result;
7222         },
7223         set: function( tween ) {
7224
7225             // Use step hook for back compat.
7226             // Use cssHook if its there.
7227             // Use .style if available and use plain properties where available.
7228             if ( jQuery.fx.step[ tween.prop ] ) {
7229                 jQuery.fx.step[ tween.prop ]( tween );
7230             } else if ( tween.elem.nodeType === 1 && (
7231                 jQuery.cssHooks[ tween.prop ] ||
7232                 tween.elem.style[ finalPropName( tween.prop ) ] != null ) ) {
7233                 jQuery.style( tween.elem, tween.prop, tween.now + tween.unit );
7234             } else {
7235                 tween.elem[ tween.prop ] = tween.now;
7236             }
7237         }
7238     },
7239 };
7240
7241 // Support: IE <=9 only
7242 // Panic based approach to setting things on disconnected nodes
7243 Tween.propHooks.scrollTop = Tween.propHooks.scrollLeft = {
7244     set: function( tween ) {
7245         if ( tween.elem.nodeType && tween.elem.parentNode ) {
7246             tween.elem[ tween.prop ] = tween.now;
7247         }
7248     }
7249 };
7250
7251 jQuery.easing = {
7252     linear: function( p ) {
7253         return p;
7254     },
7255     swing: function( p ) {
7256         return 0.5 - Math.cos( p * Math.PI ) / 2;
7257     },
7258     _default: "swing"
7259 };
7260
7261 jQuery.fx = Tween.prototype.init;
7262
7263 // Back compat <1.8 extension point
7264 jQuery.fx.step = {};
7265
7266
7267
7268
7269 var
7270     fxNow, inProgress,
7271     rfxTypes = /^(?:toggle|show|hide)$/i,
7272     rrun = /^queueHooks$/i;
7273
7274 function schedule() {
7275     if ( inProgress ) {
7276         if ( document.hidden === false && window.requestAnimationFrame ) {

```

```

7277     window.requestAnimationFrame( schedule );
7278 } else {
7279     window.setTimeout( schedule, jQuery.fx.interval );
7280 }
7281
7282 jQuery.fx.tick();
7283 }
7284 }
7285
7286 // Animations created synchronously will run synchronously
7287 function createFxNow() {
7288     window.setTimeout( function() {
7289         fxNow = undefined;
7290     } );
7291     return ( fxNow = Date.now() );
7292 }
7293
7294 // Generate parameters to create a standard animation
7295 function genFx( type, includeWidth ) {
7296     var which,
7297         i = 0,
7298         attrs = { height: type };
7299
7300     // If we include width, step value is 1 to do all cssExpand values,
7301     // otherwise step value is 2 to skip over Left and Right
7302     includeWidth = includeWidth ? 1 : 0;
7303     for ( ; i < 4; i += 2 - includeWidth ) {
7304         which = cssExpand[ i ];
7305         attrs[ "margin" + which ] = attrs[ "padding" + which ] = type;
7306     }
7307
7308     if ( includeWidth ) {
7309         attrs.opacity = attrs.width = type;
7310     }
7311
7312     return attrs;
7313 }
7314
7315 function createTween( value, prop, animation ) {
7316     var tween,
7317         collection = ( Animation.tweeners[ prop ] || [] ).concat( Animation.tweeners[ "*" ] ),
7318         index = 0,
7319         length = collection.length;
7320     for ( ; index < length; index++ ) {
7321         if ( ( tween = collection[ index ].call( animation, prop, value ) ) ) {
7322
7323             // We're done with this property
7324             return tween;
7325         }
7326     }
7327 }
7328
7329 function defaultPrefilter( elem, props, opts ) {
7330     var prop, value, toggle, hooks, oldfire, propTween, restoreDisplay, display,
7331         isBox = "width" in props || "height" in props,
7332         anim = this,
7333         orig = {},
7334         style = elem.style,
7335         hidden = elem.nodeType && isHiddenWithinTree( elem ),
7336         dataShow = dataPriv.get( elem, "fxshow" ),
7337
7338         // Queue-skipping animations hijack the fx hooks
7339         if ( !opts.queue ) {
7340             hooks = jQuery._queueHooks( elem, "fx" );
7341             if ( hooks.unqueued == null ) {
7342                 hooks.unqueued = 0;
7343                 oldfire = hooks.empty.fire;
7344                 hooks.empty.fire = function() {
7345                     if ( !hooks.unqueued ) {
7346                         oldfire();
7347                     }
7348                 };
7349             }
7350             hooks.unqueued++;
7351
7352             anim.always( function() {
7353                 // Ensure the complete handler is called before this completes
7354                 anim.always( function() {
7355                     hooks.unqueued--;
7356                     if ( !jQuery.queue( elem, "fx" ).length ) {
7357                         hooks.empty.fire();
7358                     }
7359                 } );
7360             } );
7361         };
7362     }
7363
7364     // Detect show/hide animations
7365     for ( prop in props ) {
7366         value = props[ prop ];
7367         if ( rfxTypes.test( value ) ) {
7368             delete props[ prop ];
7369             toggle = toggle || value === "toggle";
7370             if ( value === ( hidden ? "hide" : "show" ) ) {
7371
7372                 // Pretend to be hidden if this is a "show" and
7373                 // there is still data from a stopped show/hide
7374                 if ( value === "show" && dataShow && dataShow[ prop ] !== undefined ) {
7375                     hidden = true;
7376                 }
7377
7378                 // Ignore all other no-op show/hide data
7379             } else {
7380                 continue;
7381             }
7382             orig[ prop ] = dataShow && dataShow[ prop ] || jQuery.style( elem, prop );
7383         }
7384     }
7385
7386     // Bail out if this is a no-op like .hide().hide()
7387     propTween = !jQuery.isEmptyObject( props );
7388     if ( !propTween && jQuery.isEmptyObject( orig ) ) {
7389         return;
7390     }
7391

```

```

7392 // Restrict "overflow" and "display" styles during box animations
7393 if ( isBox && elem.nodeType === 1 ) {
7394
7395     // Support: IE <=9 - 11, Edge 12 - 15
7396     // Record all 3 overflow attributes because IE does not infer the shorthand
7397     // from identically-valued overflowX and overflowY and Edge just mirrors
7398     // the overflowX value there.
7399     opts.overflow = { style: overflow, styleOverflowX: style.overflowX, styleOverflowY: style.overflowY };
7400
7401     // Identify a display type, preferring old show/hide data over the CSS cascade
7402     restoreDisplay = dataShow && dataShow.display;
7403     if ( restoreDisplay == null ) {
7404         restoreDisplay = dataPriv.get( elem, "display" );
7405     }
7406     display = jQuery.css( elem, "display" );
7407     if ( display === "none" ) {
7408         if ( restoreDisplay ) {
7409             display = restoreDisplay;
7410         } else {
7411             // Get nonempty value(s) by temporarily forcing visibility
7412             showHide( { elem }, true );
7413             restoreDisplay = elem.style.display || restoreDisplay;
7414             display = jQuery.css( elem, "display" );
7415             showHide( { elem } );
7416         }
7417     }
7418 }
7419
7420 // Animate inline elements as inline-block
7421 if ( display === "inline" || display === "inline-block" && restoreDisplay != null ) {
7422     if ( jQuery.css( elem, "float" ) === "none" ) {
7423
7424         // Restore the original display value at the end of pure show/hide animations
7425         if ( !propTween ) {
7426             anim.done( function() {
7427                 style.display = restoreDisplay;
7428             } );
7429         } else {
7430             if ( restoreDisplay == null ) {
7431                 display = style.display;
7432                 restoreDisplay = display === "none" ? "" : display;
7433             }
7434             style.display = "inline-block";
7435         }
7436     }
7437 }
7438
7439 if ( opts.overflow ) {
7440     style.overflow = "hidden";
7441     anim.always( function() {
7442         style.overflow = opts.overflow[ 0 ];
7443         style.overflowX = opts.overflow[ 1 ];
7444         style.overflowY = opts.overflow[ 2 ];
7445     } );
7446 }
7447
7448 // Implement show/hide animations
7449 propTween = false;
7450 for ( prop in orig ) {
7451
7452     // General show/hide setup for this element animation
7453     if ( !propTween ) {
7454         if ( dataShow ) {
7455             if ( "hidden" in dataShow ) {
7456                 hidden = dataShow.hidden;
7457             }
7458         } else {
7459             dataShow = dataPriv.access( elem, "fxshow", { display: restoreDisplay } );
7460         }
7461
7462         // Store hidden/visible for toggle so `.stop().toggle()` "reverses"
7463         if ( toggle ) {
7464             dataShow.hidden = hidden;
7465         }
7466
7467         // Show elements before animating them
7468         if ( hidden ) {
7469             showHide( { elem }, true );
7470         }
7471     }
7472     /* eslint-disable no-loop-func */
7473     anim.done( function() {
7474         /* eslint-enable no-loop-func */
7475
7476         // The final step of a "hide" animation is actually hiding the element
7477         if ( !hidden ) {
7478             showHide( { elem } );
7479         }
7480         dataPriv.remove( elem, "fxshow" );
7481         for ( prop in orig ) {
7482             jQuery.style( elem, prop, orig[ prop ] );
7483         }
7484     } );
7485 }
7486
7487 // Per-property setup
7488 propTween = createTween( hidden ? dataShow[ prop ] : 0, prop, anim );
7489 if ( !dataShow[ prop ] ) {
7490     dataShow[ prop ] = propTween.start;
7491 } else {
7492     propTween.end = propTween.start;
7493     propTween.start = 0;
7494 }
7495
7496 function propFilter( props, specialEasing ) {
7497     var index, name, easing, value, hooks;
7498
7499     // camelCase, specialEasing and expand cssHook pass
7500     for ( index in props ) {
7501         name = camelCase( index );

```



```

7507 easing = specialEasing[ name ];
7508 value = props[ index ];
7509 if ( Array.isArray( value ) ) {
7510   easing = value[ 1 ];
7511   value = props[ index ] = value[ 0 ];
7512 }
7513
7514 if ( index !== name ) {
7515   props[ name ] = value;
7516   delete props[ index ];
7517 }
7518
7519 hooks = jQuery.cssHooks[ name ];
7520 if ( hooks && "expand" in hooks ) {
7521   value = hooks.expand( value );
7522   delete props[ name ];
7523 }
7524 // Not quite $.extend, this won't overwrite existing keys.
7525 // Reusing 'index' because we have the correct "name"
7526 for ( index in value ) {
7527   if ( !( index in props ) ) {
7528     props[ index ] = value[ index ];
7529     specialEasing[ index ] = easing;
7530   }
7531 }
7532 } else {
7533   specialEasing[ name ] = easing;
7534 }
7535 }
7536 }
7537
7538 function Animation( elem, properties, options ) {
7539   var result,
7540       stopped,
7541       index = 0,
7542       length = Animation.prefilters.length,
7543       deferred = jQuery.Deferred().always( function() {
7544
7545         // Don't match elem in the :animated selector
7546         delete tick.elem;
7547       } ),
7548       tick = function() {
7549         if ( stopped ) {
7550           return false;
7551         }
7552         var currentTime = fxNow || createFxNow(),
7553             remaining = Math.max( 0, animation.startTime + animation.duration - currentTime ),
7554
7555             // Support: Android 2.3 only
7556             // Archaic crash bug won't allow us to use `1 - ( 0.5 || 0 )` (#12497)
7557             temp = remaining / animation.duration || 0,
7558             percent = 1 - temp,
7559             index = 0,
7560             length = animation.tweens.length;
7561
7562         for ( ; index < length; index++ ) {
7563           animation.tweens[ index ].run( percent );
7564         }
7565
7566         deferred.notifyWith( elem, [ animation, percent, remaining ] );
7567
7568         // If there's more to do, yield
7569         if ( percent < 1 && length ) {
7570           return remaining;
7571         }
7572
7573         // If this was an empty animation, synthesize a final progress notification
7574         if ( !length ) {
7575           deferred.notifyWith( elem, [ animation, 1, 0 ] );
7576         }
7577
7578         // Resolve the animation and report its conclusion
7579         deferred.resolveWith( elem, [ animation ] );
7580         return false;
7581       },
7582       animation = deferred.promise( {
7583         elem: elem,
7584         props: jQuery.extend( {}, properties ),
7585         opts: jQuery.extend( true, {
7586           specialEasing: {},
7587           easing: jQuery.easing._default
7588         }, options ),
7589         originalProperties: properties,
7590         originalOptions: options,
7591         startTime: fxNow || createFxNow(),
7592         duration: options.duration,
7593         tweens: [],
7594         createTween: function( prop, end ) {
7595           var tween = jQuery.Tween( elem, animation.opts, prop, end,
7596               animation.opts.specialEasing[ prop ] || animation.opts.easing );
7597           animation.tweens.push( tween );
7598           return tween;
7599         },
7600         stop: function( gotoEnd ) {
7601           var index = 0,
7602
7603               // If we are going to the end, we want to run all the tweens
7604               // otherwise we skip this part
7605               length = gotoEnd ? animation.tweens.length : 0;
7606           if ( stopped ) {
7607             return this;
7608           }
7609           stopped = true;
7610           for ( ; index < length; index++ ) {
7611             animation.tweens[ index ].run( 1 );
7612           }
7613
7614           // Resolve when we played the last frame; otherwise, reject
7615           if ( gotoEnd ) {
7616             deferred.notifyWith( elem, [ animation, 1, 0 ] );
7617             deferred.resolveWith( elem, [ animation, gotoEnd ] );
7618           } else {
7619             deferred.rejectWith( elem, [ animation, gotoEnd ] );
7620           }
7621           return this;

```

```

7622     }
7623   } },
7624   props = animation.props,
7625
7626   propFilter| props, animation.opts.specialEasing );
7627
7628   for ( ; index < length; index++ ) {
7629     result = Animation.prefilters| index | call| animation, elem, props, animation.opts );
7630     if ( result ) {
7631       if ( !isFunction( result.stop ) ) {
7632         jQuery.queueHooks| animation.elem, animation.opts.queue | stop =
7633           result.stop.bind( result );
7634       }
7635       return result;
7636     }
7637   }
7638
7639   jQuery.map| props, createTween, animation );
7640
7641   if ( !isFunction| animation.opts.start ) ) {
7642     animation.opts.start.call( elem, animation );
7643   }
7644
7645   // Attach callbacks from options
7646   animation
7647     .progress| animation.opts.progress )
7648     .done| animation.opts.done, animation.opts.complete )
7649     .fail| animation.opts.fail )
7650     .always| animation.opts.always );
7651
7652   jQuery.fx.timer(
7653     jQuery.extend| tick, {
7654       elem: elem,
7655       anim: animation,
7656       queue: animation.opts.queue
7657     } )
7658   );
7659
7660   return animation;
7661 }
7662
7663 jQuery.Animation = jQuery.extend| Animation, {
7664
7665   tweeners: {
7666     "*": function| prop, value ) {
7667       var tween = this.createTween| prop, value );
7668       adjustCSS| tween.elem, prop, rcssNum.exec| value ), tween );
7669       return tween;
7670     }
7671   },
7672
7673   tweener: function| props, callback ) {
7674     if ( !isFunction| props ) {
7675       callback = props;
7676       props = [ "*" ];
7677     } else {
7678       props = props.match| rnothtmlwhite );
7679     }
7680
7681     var prop,
7682         index = 0,
7683         length = props.length;
7684
7685     for ( ; index < length; index++ ) {
7686       prop = props[ index ];
7687       Animation.tweeners| prop ] = Animation.tweeners| prop ] || [];
7688       Animation.tweeners| prop | unshift| callback );
7689     }
7690   },
7691
7692   prefilters: [ defaultPrefilter ],
7693
7694   prefilter: function| callback, prepend ) {
7695     if ( !prepend ) {
7696       Animation.prefilters.unshift| callback );
7697     } else {
7698       Animation.prefilters.push| callback );
7699     }
7700   }
7701 };
7702
7703 jQuery.speed = function| speed, easing, fn ) {
7704   var opt = speed && typeof speed !== "object" ? jQuery.extend| {}, speed ) : {
7705     complete: fn || !fn && easing ||
7706       isFunction| speed ) && speed,
7707     duration: speed,
7708     easing: fn && easing || easing && isFunction| easing ) && easing
7709   };
7710
7711   // Go to the end state if fx are off
7712   if ( !jQuery.fx.off ) {
7713     opt.duration = 0;
7714   }
7715   } else {
7716     if ( typeof opt.duration !== "number" ) {
7717       if ( !opt.duration in jQuery.fx.speeds ) {
7718         opt.duration = jQuery.fx.speeds.opt.duration );
7719       }
7720     } else {
7721       opt.duration = jQuery.fx.speeds._default;
7722     }
7723   }
7724
7725   // Normalize opt.queue - true/undefined/null -> "fx"
7726   if ( !opt.queue || null || opt.queue === true ) {
7727     opt.queue = "fx";
7728   }
7729
7730   // Queueing
7731   opt.old = opt.complete;
7732
7733   opt.complete = function| {
7734     if ( !isFunction| opt.old ) {
7735       opt.old.call| this );
7736     }

```

```

7737     }
7738
7739     if ( opt.queue ) {
7740         jQuery.dequeue( this, opt.queue );
7741     }
7742 };
7743
7744 return opt;
7745 };
7746
7747 jQuery.fn.extend( {
7748     fadeIn: function( speed, to, easing, callback ) {
7749
7750         // Show any hidden elements after setting opacity to 0
7751         return this.filter( isHiddenWithinTree ).css( "opacity", 0 ).show();
7752
7753         // Animate to the value specified
7754         end().animate( { opacity: to }, speed, easing, callback );
7755     },
7756     animate: function( prop, speed, easing, callback ) {
7757         var empty = jQuery.isEmptyObject( prop ),
7758             optall = jQuery.speed( speed, easing, callback ),
7759             doAnimation = function() {
7760
7761                 // Operate on a copy of prop so per-property easing won't be lost
7762                 var anim = Animation( this, jQuery.extend( {}, prop ), optall );
7763
7764                 // Empty animations, or finishing resolves immediately
7765                 if ( empty || dataPriv.get( this, "finish" ) ) {
7766                     anim.stop( true );
7767                 }
7768             };
7769
7770             doAnimation.finish = doAnimation;
7771
7772             return empty || optall.queue === false ?
7773                 this.each( doAnimation ) :
7774                 this.queue( optall.queue, doAnimation );
7775         },
7776         stop: function( type, clearQueue, gotoEnd ) {
7777             var stopQueue = function( hooks ) {
7778                 var stop = hooks.stop;
7779                 delete hooks.stop;
7780                 stop( gotoEnd );
7781             };
7782
7783             if ( typeof type !== "string" ) {
7784                 gotoEnd = clearQueue;
7785                 clearQueue = type;
7786                 type = undefined;
7787             }
7788             if ( clearQueue ) {
7789                 this.queue( type || "fx", [] );
7790             }
7791
7792             return this.each( function() {
7793                 var dequeue = true,
7794                     index = type != null ? type + "queueHooks" :
7795                     jQuery.timers,
7796                     data = dataPriv.get( this );
7797
7798                 if ( index ) {
7799                     if ( data[ index ] && data[ index ].stop ) {
7800                         stopQueue( data[ index ] );
7801                     }
7802                 } else {
7803                     for ( index in data ) {
7804                         if ( data[ index ] && data[ index ].stop && runTest( index ) ) {
7805                             stopQueue( data[ index ] );
7806                         }
7807                     }
7808                 }
7809
7810                 for ( index = timers.length; index--; ) {
7811                     if ( timers[ index ].elem === this &&
7812                         ( type == null || timers[ index ].queue === type ) ) {
7813
7814                         timers[ index ].anim.stop( gotoEnd );
7815                         dequeue = false;
7816                         timers.splice( index, 1 );
7817                     }
7818                 }
7819
7820                 // Start the next in the queue if the last step wasn't forced.
7821                 // Timers currently will call their complete callbacks, which
7822                 // will dequeue but only if they were gotoEnd.
7823                 if ( dequeue || !gotoEnd ) {
7824                     jQuery.dequeue( this, type );
7825                 }
7826             } );
7827         },
7828         finish: function( type ) {
7829             if ( type !== false ) {
7830                 type = type || "fx";
7831             }
7832             return this.each( function() {
7833                 var index,
7834                     data = dataPriv.get( this ),
7835                     queue = data[ type + "queue" ],
7836                     hooks = data[ type + "queueHooks" ],
7837                     timers = jQuery.timers,
7838                     length = queue ? queue.length : 0;
7839
7840                 // Enable finishing flag on private data
7841                 data.finish = true;
7842
7843                 // Empty the queue first
7844                 jQuery.queue( this, type, [] );
7845
7846                 if ( hooks && hooks.stop ) {
7847                     hooks.stop.call( this, true );
7848                 }
7849
7850                 // Look for any active animations, and finish them
7851                 for ( index = timers.length; index--; ) {

```

```

7852     if ( timers[ index ] elem === this && timers[ index ] queue === type ) {
7853         timers[ index ] anim stop( true );
7854         timers.splice( index, 1 );
7855     }
7856 }
7857
7858 // Look for any animations in the old queue and finish them
7859 for ( index = 0; index < length; index++ ) {
7860     if ( queue[ index ] && queue[ index ].finish ) {
7861         queue[ index ].finish.call( this );
7862     }
7863 }
7864
7865 // Turn off finishing flag
7866 delete data.finish;
7867 } };
7868 }
7869 } );
7870
7871 jQuery.each( [ "toggle", "show", "hide" ], function( _i, name ) {
7872     var cssFn = jQuery.fn[ name ];
7873     jQuery.fn[ name ] = function( speed, easing, callback ) {
7874         return speed == null || typeof speed === "boolean" ?
7875             cssFn.apply( this, arguments ) :
7876             this.animate( genFx( name, true ), speed, easing, callback );
7877     };
7878 } );
7879
7880 // Generate shortcuts for custom animations
7881 jQuery.each( {
7882     slideDown: genFx( "show" ),
7883     slideUp: genFx( "hide" ),
7884     slideToggle: genFx( "toggle" ),
7885     fadeIn: { opacity: "show" },
7886     fadeOut: { opacity: "hide" },
7887     fadeToggle: { opacity: "toggle" }
7888 }, function( name, props ) {
7889     jQuery.fn[ name ] = function( speed, easing, callback ) {
7890         return this.animate( props, speed, easing, callback );
7891     };
7892 } );
7893
7894 jQuery.timers = [];
7895 jQuery.fx.tick = function() {
7896     var timer;
7897     i = 0;
7898     timers = jQuery.timers;
7899
7900     fxNow = Date.now();
7901
7902     for ( ; i < timers.length; i++ ) {
7903         timer = timers[ i ];
7904
7905         // Run the timer and safely remove it when done (allowing for external removal)
7906         if ( !timer() && timers[ i ] === timer ) {
7907             timers.splice( i--, 1 );
7908         }
7909     }
7910
7911     if ( !timers.length ) {
7912         jQuery.fx.stop();
7913     }
7914     fxNow = undefined;
7915 };
7916
7917 jQuery.fx.timer = function( timer ) {
7918     jQuery.timers.push( timer );
7919     jQuery.fx.start();
7920 };
7921
7922 jQuery.fx.interval = 13;
7923 jQuery.fx.start = function() {
7924     if ( !inProgress ) {
7925         return;
7926     }
7927
7928     inProgress = true;
7929     schedule();
7930 };
7931
7932 jQuery.fx.stop = function() {
7933     inProgress = null;
7934 };
7935
7936 jQuery.fx.speeds = {
7937     slow: 600,
7938     fast: 200,
7939
7940     // Default speed
7941     _default: 400
7942 };
7943
7944
7945 // Based off of the plugin by Clint Helfers, with permission.
7946 // https://web.archive.org/web/20100324014747/http://blindsignals.com/index.php/2009/07/jquery-delay/
7947 jQuery.fn.delay = function( time, type ) {
7948     time = jQuery.fx ? jQuery.fx.speeds[ time ] || time : time;
7949     type = type || "fx";
7950
7951     return this.queue( type, function( next, hooks ) {
7952         var timeout = window.setTimeout( next, time );
7953         hooks.stop = function() {
7954             window.clearTimeout( timeout );
7955         };
7956     } );
7957 };
7958
7959
7960 ( function() {
7961     var input = document.createElement( "input" ),
7962         select = document.createElement( "select" ),
7963         opt = select.appendChild( document.createElement( "option" ) );
7964
7965     input.type = "checkbox";
7966

```

```

7967 // Support: Android <=4.3 only
7968 // Default value for a checkbox should be "on"
7969 support checkOn = input value !== "";
7970
7971 // Support: IE <=11 only
7972 // Must access selectedIndex to make default options select
7973 support optSelected = opt.selected;
7974
7975 // Support: IE <=11 only
7976 // An input loses its value after becoming a radio
7977 input = document.createElement( "input" );
7978 input.value = "t";
7979 input.type = "radio";
7980 support radioValue = input.value === "t";
7981 }());
7982
7983
7984 var boolHook,
7985 attrHandle = jQuery.expr.attrHandle;
7986
7987 jQuery.fn.extend( {
7988   attr: function( name, value ) {
7989     return access( this, jQuery.attr, name, value, arguments.length > 1 );
7990   },
7991   removeAttr: function( name ) {
7992     return this.each( function() {
7993       jQuery.removeAttr( this, name );
7994     } );
7995   }
7996 });
7997
7998 jQuery.extend( {
7999   attr: function( elem, name, value ) {
8000     var ret, hooks;
8001     nType = elem.nodeType;
8002
8003     // Don't get/set attributes on text, comment and attribute nodes
8004     if ( nType === 3 || nType === 8 || nType === 2 ) {
8005       return;
8006     }
8007
8008     // Fallback to prop when attributes are not supported
8009     if ( typeof elem.getAttribute !== "undefined" ) {
8010       return jQuery.prop( elem, name, value );
8011     }
8012
8013     // Attribute hooks are determined by the lowercase version
8014     // Grab necessary hook if one is defined
8015     if ( nType !== 1 || !jQuery.isXMLDoc( elem ) ) {
8016       hooks = jQuery.attrHooks[ name.toLowerCase() ];
8017       ( jQuery.expr.match.bool.test( name ) ? boolHook : undefined );
8018     }
8019
8020     if ( value !== undefined ) {
8021       if ( value === null ) {
8022         jQuery.removeAttr( elem, name );
8023         return;
8024       }
8025
8026       if ( hooks && "set" in hooks &&
8027         ( ret = hooks.set( elem, value, name ) ) !== undefined ) {
8028         return ret;
8029       }
8030
8031       elem.setAttribute( name, value + "" );
8032       return value;
8033     }
8034
8035     if ( hooks && "get" in hooks && ( ret = hooks.get( elem, name ) ) !== null ) {
8036       return ret;
8037     }
8038
8039     ret = jQuery.find.attr( elem, name );
8040
8041     // Non-existent attributes return null, we normalize to undefined
8042     return ret === null ? undefined : ret;
8043   },
8044   attrHooks: {
8045     type: {
8046       set: function( elem, value ) {
8047         if ( !support.radioValue && value === "radio" &&
8048           nodeName( elem, "input" ) ) {
8049           var val = elem.value;
8050           elem.setAttribute( "type", value );
8051           if ( val ) {
8052             elem.value = val;
8053           }
8054         }
8055         return value;
8056       }
8057     }
8058   },
8059   },
8060   },
8061   removeAttr: function( elem, value ) {
8062     var name,
8063         i = 0,
8064         attrNames = value && value.match( rnothtmlwhite );
8065
8066     // Attribute names can contain non-HTML whitespace characters
8067     // https://html.spec.whatwg.org/multipage/syntax.html#attributes-2
8068     attrNames = value && value.match( rnothtmlwhite );
8069
8070     if ( attrNames && elem.nodeType === 1 ) {
8071       while ( ( name = attrNames[ i++ ] ) ) {
8072         elem.removeAttribute( name );
8073       }
8074     }
8075   }
8076 });
8077
8078 // Hooks for boolean attributes
8079 boolHook = {
8080   set: function( elem, value, name ) {
8081     if ( value === false ) {

```

```

8082
8083 // Remove boolean attributes when set to false
8084 jQuery.removeAttr( elem, name );
8085 } else {
8086     elem.setAttribute( name, name );
8087 }
8088 return name;
8089 }
8090 };
8091
8092 jQuery.each( jQuery.expr.match.bool.source.match( /\w+/g ), function( _i, name ) {
8093     var getter = attrHandle[ name ] || jQuery.find.attr;
8094
8095     attrHandle[ name ] = function( elem, name, isXML ) {
8096         var ret, handle;
8097         lowercaseName = name.toLowerCase();
8098
8099         if ( !isXML ) {
8100             // Avoid an infinite loop by temporarily removing this function from the getter
8101             handle = attrHandle[ lowercaseName ];
8102             attrHandle[ lowercaseName ] = ret;
8103             ret = getter( elem, name, isXML ) != null ?
8104                 lowercaseName :
8105                 null;
8106             attrHandle[ lowercaseName ] = handle;
8107         }
8108         return ret;
8109     };
8110 });
8111
8112
8113
8114
8115
8116 var rfocusable = /^(?:input|select|textarea|button)$/i;
8117 rclickable = /^(?:a|area)$/i;
8118
8119 jQuery.fn.extend( {
8120     prop: function( name, value ) {
8121         return access( this, jQuery.prop, name, value, arguments.length > 1 );
8122     },
8123
8124     removeProp: function( name ) {
8125         return this.each( function() {
8126             delete this[ jQuery.propFix[ name ] || name ];
8127         } );
8128     },
8129 });
8130
8131 jQuery.extend( {
8132     prop: function( elem, name, value ) {
8133         var ret, hooks;
8134         nType = elem.nodeType;
8135
8136         // Don't get/set properties on text, comment and attribute nodes
8137         if ( nType === 3 || nType === 8 || nType === 2 ) {
8138             return;
8139         }
8140
8141         if ( nType !== 1 || !jQuery.isXMLDoc( elem ) ) {
8142             // Fix name and attach hooks
8143             name = jQuery.propFix[ name ] || name;
8144             hooks = jQuery.propHooks[ name ];
8145
8146             if ( value !== undefined ) {
8147                 if ( hooks && "set" in hooks &&
8148                     ( ret = hooks.set( elem, value, name ) ) !== undefined ) {
8149                     return ret;
8150                 }
8151                 return ( elem[ name ] = value );
8152             }
8153
8154             if ( hooks && "get" in hooks && ( ret = hooks.get( elem, name ) ) !== null ) {
8155                 return ret;
8156             }
8157
8158             return elem[ name ];
8159         },
8160
8161         propHooks: {
8162             tabIndex: {
8163                 get: function( elem ) {
8164                     // Support: IE <= 9 - 11 only
8165                     // elem.tabIndex doesn't always return the
8166                     // correct value when it hasn't been explicitly set
8167                     // https://web.archive.org/web/20141116233347/http://fluidproject.org/blog/2008/01/09/getting-setting-and-removing-tabindex-values-with-javascript/
8168                     // Use proper attribute retrieval (#12072)
8169                     var tabindex = jQuery.find.attr( elem, "tabindex" );
8170
8171                     if ( tabindex ) {
8172                         return parseInt( tabindex, 10 );
8173                     }
8174
8175                     if (
8176                         rfocusable.test( elem.nodeName ) ||
8177                         rclickable.test( elem.nodeName ) &&
8178                         elem.href
8179                     ) {
8180                         return 0;
8181                     }
8182
8183                     return -1;
8184                 }
8185             },
8186
8187             propFix: {
8188                 "for": "htmlFor",
8189                 "class": "className"
8190             }
8191         }
8192     });
8193
8194
8195
8196

```

```

8197
8198 // Support: IE <=11 only
8199 // Accessing the selectedIndex property
8200 // forces the browser to respect setting selected
8201 // on the option
8202 // The getter ensures a default option is selected
8203 // when in an optgroup
8204 // eslint rule "no-unused-expressions" is disabled for this code
8205 // since it considers such accessions noop
8206 if ( !support.optSelected ) {
8207   jQuery.propHooks.selected = {
8208     get: function( elem ) {
8209
8210       /* eslint no-unused-expressions: "off" */
8211
8212       var parent = elem.parentNode;
8213       if ( parent && parent.parentNode ) {
8214         parent.parentNode.selectedIndex;
8215       }
8216       return null;
8217     },
8218     set: function( elem ) {
8219
8220       /* eslint no-unused-expressions: "off" */
8221
8222       var parent = elem.parentNode;
8223       if ( parent ) {
8224         parent.selectedIndex;
8225
8226         if ( parent.parentNode ) {
8227           parent.parentNode.selectedIndex;
8228         }
8229       }
8230     };
8231   };
8232 }
8233
8234 jQuery.each( [
8235   "tabIndex",
8236   "readOnly",
8237   "maxLength",
8238   "cellSpacing",
8239   "cellPadding",
8240   "rowSpan",
8241   "colSpan",
8242   "useMap",
8243   "frameBorder",
8244   "contentEditable"
8245 ], function() {
8246   jQuery.propFix[ this.toLowerCase() ] = this;
8247 });
8248
8249
8250
8251
8252 // Strip and collapse whitespace according to HTML spec
8253 // https://infra.spec.whatwg.org/#strip-and-collapse-ascii-whitespace
8254 function stripAndCollapse( value ) {
8255   var tokens = value.match( rnohtmlwhite ) || [];
8256   return tokens.join( " " );
8257 }
8258
8259
8260 function getClass( elem ) {
8261   return elem.getAttribute( && elem.getAttribute( "class" ) || "" );
8262 }
8263
8264 function classesToArray( value ) {
8265   if ( Array.isArray( value ) ) {
8266     return value;
8267   }
8268   if ( typeof value === "string" ) {
8269     return value.match( rnohtmlwhite ) || [];
8270   }
8271   return [];
8272 }
8273
8274 jQuery.fn.extend( {
8275   addClass: function( value ) {
8276     var classes = elem, cur, curValue, clazz, j, finalValue,
8277       i = 0;
8278
8279     if ( isFunction( value ) ) {
8280       return this.each( function( j ) {
8281         jQuery( this ).addClass( value.call( this, j, getClass( this ) ) );
8282       } );
8283     }
8284
8285     classes = classesToArray( value );
8286
8287     if ( classes.length ) {
8288       while ( ( elem = this[ i++ ] ) ) {
8289         curValue = getClass( elem );
8290         cur = elem.nodeType === 1 && ( " " + stripAndCollapse( curValue ) + " " );
8291
8292         if ( cur ) {
8293           j = 0;
8294           while ( ( clazz = classes[ j++ ] ) ) {
8295             if ( cur.indexOf( " " + clazz + " " ) < 0 ) {
8296               cur += clazz + " ";
8297             }
8298           }
8299
8300           // Only assign if different to avoid unneeded rendering.
8301           finalValue = stripAndCollapse( cur );
8302           if ( curValue !== finalValue ) {
8303             elem.setAttribute( "class", finalValue );
8304           }
8305         }
8306       }
8307     }
8308
8309     return this;
8310   },
8311

```

```

8312 removeClass: function( value ) {
8313     var classes, elem, cur, curValue, clazz, j, finalValue,
8314         i = 0;
8315     if ( !isFunction( value ) ) {
8316         return this.each( function( j ) {
8317             jQuery( this ).removeClass( value ).call( this, j, getClass( this ) );
8318         } );
8319     }
8320     if ( !arguments.length ) {
8321         return this.attr( "class", "" );
8322     }
8323     classes = classesToArray( value );
8324     if ( classes.length ) {
8325         while ( ( elem = this[ i++ ] ) ) {
8326             curValue = getClass( elem );
8327             // This expression is here for better compressibility (see addClass)
8328             cur = elem.nodeType === 1 && ( " " + stripAndCollapse( curValue ) + " " );
8329             if ( cur ) {
8330                 // Remove *all* instances
8331                 while ( ( clazz = classes[ j++ ] ) ) {
8332                     cur = cur.replace( " " + clazz + " ", "" );
8333                 }
8334             }
8335             // Only assign if different to avoid unneeded rendering.
8336             finalValue = stripAndCollapse( cur );
8337             if ( curValue !== finalValue ) {
8338                 elem.setAttribute( "class", finalValue );
8339             }
8340         }
8341     }
8342     return this;
8343 },
8344 toggleClass: function( value, stateVal ) {
8345     var type = typeof value,
8346         isValidValue = type === "string" || Array.isArray( value );
8347     if ( type === "boolean" && !isValidValue ) {
8348         return stateVal ? this.addClass( value ) : this.removeClass( value );
8349     }
8350     if ( !isFunction( value ) ) {
8351         return this.each( function( i ) {
8352             jQuery( this ).toggleClass(
8353                 value.call( this, i, getClass( this ), stateVal ),
8354                 stateVal
8355             );
8356         } );
8357     }
8358     return this.each( function() {
8359         var className, i, self, classNames;
8360         if ( !isValidValue ) {
8361             // Toggle individual class names
8362             i = 0;
8363             self = jQuery( this );
8364             classNames = classesToArray( value );
8365             while ( ( className = classNames[ i++ ] ) ) {
8366                 // Check each className given, space separated list
8367                 if ( self.hasClass( className ) ) {
8368                     self.removeClass( className );
8369                 } else {
8370                     self.addClass( className );
8371                 }
8372             }
8373             // Toggle whole class name
8374         } else if ( value === undefined || type === "boolean" ) {
8375             className = getClass( this );
8376             if ( className ) {
8377                 // Store className if set
8378                 dataPriv.set( this, "__className__", className );
8379             }
8380             // If the element has a class name or if we're passed `false`,
8381             // then remove the whole classname (if there was one, the above saved it).
8382             // Otherwise bring back whatever was previously saved (if anything),
8383             // falling back to the empty string if nothing was stored.
8384             if ( this.setAttribute( "class",
8385                 className || value === false ?
8386                     "" :
8387                     dataPriv.get( this, "__className__" ) || ""
8388             );
8389             }
8390         }
8391     }
8392     return this;
8393 },
8394 hasClass: function( selector ) {
8395     var className, elem,
8396         i = 0;
8397     className = " " + selector + " ";
8398     while ( ( elem = this[ i++ ] ) ) {
8399         if ( elem.nodeType === 1 &&
8400             ( " " + stripAndCollapse( getClass( elem ) ) + " " ).indexOf( className ) > -1 ) {
8401             return true;
8402         }
8403     }
8404 }

```



```

8427     }
8428 }
8429
8430 return false;
8431 }
8432 });
8433
8434
8435
8436
8437 var rreturn = /r/g;
8438
8439 jQuery.fn.extend( {
8440   val: function( value ) {
8441     var hooks, ret, valuesFunction,
8442         elem = this[ 0 ];
8443
8444     if ( !arguments.length ) {
8445       if ( elem ) {
8446         hooks = jQuery.valHooks[ elem.type ] ||
8447           jQuery.valHooks[ elem.nodeName.toLowerCase() ];
8448
8449         if ( hooks &&
8450             "get" in hooks &&
8451             ( ret = hooks.get( elem, "value" ) ) !== undefined )
8452           {
8453             return ret;
8454           }
8455
8456         ret = elem.value;
8457
8458         // Handle most common string cases
8459         if ( typeof ret === "string" ) {
8460           return ret.replace( rreturn, "" );
8461         }
8462
8463         // Handle cases where value is null/undef or number
8464         return ret === null ? "" : ret;
8465       }
8466
8467       return;
8468     }
8469
8470     valuesFunction = isFunction( value );
8471
8472     return this.each( function( i ) {
8473       var val;
8474
8475       if ( this.nodeType !== 1 ) {
8476         return;
8477       }
8478
8479       if ( valuesFunction ) {
8480         val = value.call( this, i, jQuery( this ).val() );
8481       } else {
8482         val = value;
8483       }
8484
8485       // Treat null/undefined as ""; convert numbers to string
8486       if ( val == null ) {
8487         val = "";
8488       } else if ( typeof val === "number" ) {
8489         val += "";
8490       } else if ( !Array.isArray( val ) ) {
8491         val = jQuery.map( val, function( value ) {
8492           return value == null ? "" : value + "";
8493         } );
8494       }
8495
8496       hooks = jQuery.valHooks[ this.type ] || jQuery.valHooks[ this.nodeName.toLowerCase() ];
8497
8498       // If set returns undefined, fall back to normal setting
8499       if ( !hooks || ( !("set" in hooks) || hooks.set( this, val, "value" ) === undefined ) ) {
8500         this.value = val;
8501       }
8502     } );
8503   },
8504   set: function( value ) {
8505     return this.each( function() {
8506       this.value = value;
8507     } );
8508   },
8509   valHooks: {
8510     option: {
8511       get: function( elem ) {
8512
8513         var val = jQuery.find.attr( elem, "value" );
8514         return val != null ?
8515           val :
8516
8517           // Support: IE <= 10 - 11 only
8518           // option.text throws exceptions (#14686, #14858)
8519           // Strip and collapse whitespace
8520           // https://html.spec.whatwg.org/#strip-and-collapse-whitespace
8521           stripAndCollapse( jQuery.text( elem ) );
8522       },
8523     },
8524     select: {
8525       get: function( elem ) {
8526         var value, option, i,
8527             options = elem.options,
8528             index = elem.selectedIndex,
8529             one = elem.type === "select-one",
8530             values = one ? null : [],
8531             max = one ? index + 1 : options.length;
8532
8533         if ( index < 0 ) {
8534           i = max;
8535         } else {
8536           i = one ? index : 0;
8537         }
8538
8539         // Loop through all the selected options
8540         for ( ; i < max; i++ ) {

```

```

8542     option = options[ i ];
8543
8544     // Support: IE <=9 only
8545     // IE8-9 doesn't update selected after form reset (#2551)
8546     if ( ( option.selected || i === index ) &&
8547
8548         // Don't return options that are disabled or in a disabled optgroup
8549         ( option.disabled &&
8550           ( !option.parentNode.disabled ||
8551             !nodeName( option.parentNode, "optgroup" ) ) ) ) {
8552
8553         // Get the specific value for the option
8554         value = jQuery( option ).val();
8555
8556         // We don't need an array for one selects
8557         if ( one ) {
8558             return value;
8559         }
8560
8561         // Multi-Selects return an array
8562         values.push( value );
8563     }
8564 }
8565
8566 return values;
8567 },
8568
8569 set: function( elem, value ) {
8570     var optionSet, option,
8571         options = elem.options,
8572         values = jQuery.makeArray( value ),
8573         i = options.length;
8574
8575     while ( i-- ) {
8576         option = options[ i ];
8577
8578         /* eslint-disable no-cond-assign */
8579
8580         if ( option.selected =
8581             jQuery.inArray( jQuery.valHooks.option.get( option ), values ) > -1
8582         ) {
8583             optionSet = true;
8584         }
8585
8586         /* eslint-enable no-cond-assign */
8587     }
8588
8589     // Force browsers to behave consistently when non-matching value is set
8590     if ( !optionSet ) {
8591         elem.selectedIndex = -1;
8592     }
8593     return values;
8594 }
8595 }
8596 }
8597 }
8598
8599 // Radios and checkboxes getter/setter
8600 jQuery.each( [ "radio", "checkbox" ], function() {
8601     jQuery.valHooks[ this ] = {
8602         set: function( elem, value ) {
8603             if ( Array.isArray( value ) ) {
8604                 return ( elem.checked = jQuery.inArray( jQuery( elem ).val(), value ) > -1 );
8605             }
8606         }
8607     };
8608     if ( !support.checkOn ) {
8609         jQuery.valHooks[ this ].get = function( elem ) {
8610             return elem.getAttribute( "value" ) === null ? "on" : elem.value;
8611         };
8612     }
8613 } );
8614
8615
8616
8617
8618 // Return jQuery for attributes-only inclusion
8619
8620
8621 support.focusin = "onfocusin" in window;
8622
8623
8624 var rfocusMorph = /^(?:focusinfocus|focusoutblur)$/;
8625 stopPropagationCallback = function( e ) {
8626     e.stopPropagation();
8627 };
8628
8629 jQuery.extend( jQuery.event, {
8630
8631     trigger: function( event, data, elem, onlyHandlers ) {
8632
8633         var i, cur, tmp, bubbleType, ontype, handle, special, lastElement,
8634             eventPath = [ elem || document ],
8635             type = hasOwn.call( event, "type" ) ? event.type : event,
8636             namespaces = hasOwn.call( event, "namespace" ) ? event.namespace.split( "." ) : [];
8637
8638         cur = lastElement = tmp = elem = elem || document;
8639
8640         // Don't do events on text and comment nodes
8641         if ( elem.nodeType === 3 || elem.nodeType === 8 ) {
8642             return;
8643         }
8644
8645         // focus/blur morphs to focusin/out; ensure we're not firing them right now
8646         if ( rfocusMorph.test( type + jQuery.event.triggered ) ) {
8647             return;
8648         }
8649
8650         if ( type.indexOf( "." ) > -1 ) {
8651
8652             // Namespaced trigger; create a regexp to match event type in handle()
8653             namespaces = type.split( "." );
8654             type = namespaces.shift();
8655             namespaces.sort();
8656         }

```

```

8657 ontype = type.indexOf( ":" ) < 0 && "on" + type;
8658
8659 // Caller can pass in a jQuery.Event object, Object, or just an event type string
8660 event = event[ jQuery.expando ] ?
8661   event :
8662   new jQuery.Event( type, typeof event === "object" && event );
8663
8664 // Trigger bitmask: & 1 for native handlers; & 2 for jQuery (always true)
8665 event.isTrigger = onlyHandlers ? 2 : 3;
8666 event.namespace = namespaces.join( "." );
8667 event.namespace = event.namespace ?
8668   new RegExp( "(^|\\.)" + namespaces.join( "\\.(?:.*\\.)" ) + "(\\.|$)" ) :
8669   null;
8670
8671 // Clean up the event in case it is being reused
8672 event.result = undefined;
8673 if ( !event.target ) {
8674   event.target = elem;
8675 }
8676
8677 // Clone any incoming data and prepend the event, creating the handler arg list
8678 data = data == null ?
8679   [ event ] :
8680   jQuery.makeArray( data, [ event ] );
8681
8682 // Allow special events to draw outside the lines
8683 special = jQuery.event.special[ type ] || {};
8684 if ( !onlyHandlers && special.trigger && special.trigger.apply( elem, data ) === false ) {
8685   return;
8686 }
8687
8688 // Determine event propagation path in advance, per W3C events spec (#9951)
8689 // Bubble up to document, then to window; watch for a global ownerDocument var (#9724)
8690 if ( !onlyHandlers && !special.noBubble && !isWindow( elem ) ) {
8691
8692   bubbleType = special.delegateType || type;
8693   if ( !rfocusMorph.test( bubbleType + type ) ) {
8694     cur = cur.parentNode;
8695   }
8696   for ( ; cur; cur = cur.parentNode ) {
8697     eventPath.push( cur );
8698     tmp = cur;
8699   }
8700
8701   // Only add window if we got to document (e.g., not plain obj or detached DOM)
8702   if ( tmp === ( elem.ownerDocument || document ) ) {
8703     eventPath.push( tmp.defaultView || tmp.parentWindow || window );
8704   }
8705 }
8706
8707 // Fire handlers on the event path
8708 i = 0;
8709 while ( ( cur = eventPath[ i++ ] ) && !event.isPropagationStopped() ) {
8710   lastElement = cur;
8711   event.type = i > 1 ?
8712     bubbleType :
8713     special.bindType || type;
8714
8715   // jQuery handler
8716   handle = ( dataPriv.get( cur, "events" ) || Object.create( null ) )[ event.type ] &&
8717     dataPriv.get( cur, "handle" );
8718   if ( handle ) {
8719     handle.apply( cur, data );
8720   }
8721
8722   // Native handler
8723   handle = ontype && cur[ ontype ];
8724   if ( handle && handle.apply( cur, data ) ) {
8725     event.result = handle.apply( cur, data );
8726     if ( event.result === false ) {
8727       event.preventDefault();
8728     }
8729   }
8730 }
8731 event.type = type;
8732
8733 // If nobody prevented the default action, do it now
8734 if ( !onlyHandlers && !event.isDefaultPrevented() ) {
8735
8736   if ( !special._default ||
8737     special._default.apply( eventPath.pop(), data ) === false ) &&
8738     acceptData( elem ) ) {
8739
8740     // Call a native DOM method on the target with the same name as the event.
8741     // Don't do default actions on window, that's where global variables be (#6170)
8742     if ( ontype && isFunction( elem[ type ] ) && !isWindow( elem ) ) {
8743
8744       // Don't re-trigger an onFOO event when we call its FOO() method
8745       tmp = elem[ ontype ];
8746
8747       if ( tmp ) {
8748         elem[ ontype ] = null;
8749       }
8750
8751       // Prevent re-triggering of the same event, since we already bubbled it above
8752       jQuery.event.triggered = type;
8753
8754       if ( event.isPropagationStopped() ) {
8755         lastElement.addEventListener( type, stopPropagationCallback );
8756       }
8757
8758       elem[ type ]();
8759
8760       if ( event.isPropagationStopped() ) {
8761         lastElement.removeEventListener( type, stopPropagationCallback );
8762       }
8763
8764       jQuery.event.triggered = undefined;
8765
8766       if ( tmp ) {
8767         elem[ ontype ] = tmp;
8768       }
8769     }
8770   }
8771 }

```

```

8773     return event result;
8774 },
8775
8776 // Piggyback on a donor event to simulate a different one
8777 // Used only for "focus(in | out)" events
8778 simulate: function( type, elem, event ) {
8779     var e = jQuery.extend(
8780         new jQuery.Event(),
8781         event,
8782         {
8783             type: type,
8784             isSimulated: true
8785         }
8786     );
8787
8788     jQuery.event.trigger( e, null, elem );
8789 }
8790
8791 } );
8792
8793 jQuery.fn.extend( {
8794     trigger: function( type, data ) {
8795         return this.each( function() {
8796             jQuery.event.trigger( type, data, this );
8797         } );
8798     },
8799     triggerHandler: function( type, data ) {
8800         var elem = this[ 0 ];
8801         if ( elem ) {
8802             return jQuery.event.trigger( type, data, elem, true );
8803         }
8804     }
8805 );
8806
8807 // Support: Firefox <=44
8808 // Firefox doesn't have focus(in | out) events
8809 // Related ticket - https://bugzilla.mozilla.org/show_bug.cgi?id=687787
8810 //
8811 // Support: Chrome <=48 - 49, Safari <=9.0 - 9.1
8812 // focus(in | out) events fire after focus & blur events,
8813 // which is spec violation - http://www.w3.org/TR/DOM-Level-3-Events/#events-focusevent-order
8814 // Related ticket - https://bugs.chromium.org/p/chromium/issues/detail?id=449857
8815 if ( !support.focusin ) {
8816     jQuery.each( { focus: "focusin", blur: "focusout" }, function( orig, fix ) {
8817         // Attach a single capturing handler on the document while someone wants focusin/focusout
8818         var handler = function( event ) {
8819             jQuery.event.simulate( fix, event.target, jQuery.event.fix( event ) );
8820         };
8821
8822         jQuery.event.special[ fix ] = {
8823             setup: function() {
8824                 // Handle: regular nodes (via `this.ownerDocument`), window
8825                 // (via `this.document`) & document (via `this`).
8826                 var doc = this.ownerDocument || this.document || this,
8827                     attaches = dataPriv.access( doc, fix );
8828
8829                 if ( !attaches ) {
8830                     doc.addEventListener( orig, handler, true );
8831                 }
8832                 dataPriv.access( doc, fix, ( attaches || 0 ) + 1 );
8833             },
8834             teardown: function() {
8835                 var doc = this.ownerDocument || this.document || this,
8836                     attaches = dataPriv.access( doc, fix ) - 1;
8837
8838                 if ( !attaches ) {
8839                     doc.removeEventListener( orig, handler, true );
8840                     dataPriv.remove( doc, fix );
8841                 } else {
8842                     dataPriv.access( doc, fix, attaches );
8843                 }
8844             },
8845             add: function() {
8846                 // No longer a jQuery event - see http://bugs.jquery.com/ticket/14692
8847                 return;
8848             }
8849         };
8850     } );
8851
8852     var location = window.location;
8853     var nonce = { guid: Date.now() };
8854     var rquery = ( /\?/ );
8855
8856 // Cross-browser xml parsing
8857 jQuery.parseXML = function( data ) {
8858     var xml, parserErrorElem;
8859     if ( !data || data.nodeType === "string" ) {
8860         return null;
8861     }
8862
8863     // Support: IE 9 - 11 only
8864     // IE throws on parseFromString with invalid input.
8865     try {
8866         xml = ( new window.DOMParser() ).parseFromString( data, "text/xml" );
8867     } catch ( e ) {}
8868
8869     parserErrorElem = xml && xml.getElementsByTagName( "parsererror" )[ 0 ];
8870     if ( !xml || parserErrorElem ) {
8871         jQuery.error( "Invalid XML: " + (
8872             parserErrorElem ?
8873                 parserErrorElem.textContent :
8874                 ""
8875         ) );
8876     }
8877     return xml;
8878 }
8879
8880 // Support: IE 9 - 11 only
8881 // IE throws on parseFromString with invalid input.
8882 try {
8883     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8884 } catch ( e ) {}
8885
8886 // Support: IE 9 - 11 only
8887 // IE throws on parseFromString with invalid input.
8888 try {
8889     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8890 } catch ( e ) {}
8891
8892 // Support: IE 9 - 11 only
8893 // IE throws on parseFromString with invalid input.
8894 try {
8895     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8896 } catch ( e ) {}
8897
8898 // Support: IE 9 - 11 only
8899 // IE throws on parseFromString with invalid input.
8900 try {
8901     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8902 } catch ( e ) {}
8903
8904 // Support: IE 9 - 11 only
8905 // IE throws on parseFromString with invalid input.
8906 try {
8907     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8908 } catch ( e ) {}
8909
8910 // Support: IE 9 - 11 only
8911 // IE throws on parseFromString with invalid input.
8912 try {
8913     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8914 } catch ( e ) {}
8915
8916 // Support: IE 9 - 11 only
8917 // IE throws on parseFromString with invalid input.
8918 try {
8919     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8920 } catch ( e ) {}
8921
8922 // Support: IE 9 - 11 only
8923 // IE throws on parseFromString with invalid input.
8924 try {
8925     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8926 } catch ( e ) {}
8927
8928 // Support: IE 9 - 11 only
8929 // IE throws on parseFromString with invalid input.
8930 try {
8931     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8932 } catch ( e ) {}
8933
8934 // Support: IE 9 - 11 only
8935 // IE throws on parseFromString with invalid input.
8936 try {
8937     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8938 } catch ( e ) {}
8939
8940 // Support: IE 9 - 11 only
8941 // IE throws on parseFromString with invalid input.
8942 try {
8943     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8944 } catch ( e ) {}
8945
8946 // Support: IE 9 - 11 only
8947 // IE throws on parseFromString with invalid input.
8948 try {
8949     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8950 } catch ( e ) {}
8951
8952 // Support: IE 9 - 11 only
8953 // IE throws on parseFromString with invalid input.
8954 try {
8955     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8956 } catch ( e ) {}
8957
8958 // Support: IE 9 - 11 only
8959 // IE throws on parseFromString with invalid input.
8960 try {
8961     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8962 } catch ( e ) {}
8963
8964 // Support: IE 9 - 11 only
8965 // IE throws on parseFromString with invalid input.
8966 try {
8967     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8968 } catch ( e ) {}
8969
8970 // Support: IE 9 - 11 only
8971 // IE throws on parseFromString with invalid input.
8972 try {
8973     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8974 } catch ( e ) {}
8975
8976 // Support: IE 9 - 11 only
8977 // IE throws on parseFromString with invalid input.
8978 try {
8979     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8980 } catch ( e ) {}
8981
8982 // Support: IE 9 - 11 only
8983 // IE throws on parseFromString with invalid input.
8984 try {
8985     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8986 } catch ( e ) {}
8987
8988 // Support: IE 9 - 11 only
8989 // IE throws on parseFromString with invalid input.
8990 try {
8991     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8992 } catch ( e ) {}
8993
8994 // Support: IE 9 - 11 only
8995 // IE throws on parseFromString with invalid input.
8996 try {
8997     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
8998 } catch ( e ) {}
8999
9000 // Support: IE 9 - 11 only
9001 // IE throws on parseFromString with invalid input.
9002 try {
9003     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9004 } catch ( e ) {}
9005
9006 // Support: IE 9 - 11 only
9007 // IE throws on parseFromString with invalid input.
9008 try {
9009     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9010 } catch ( e ) {}
9011
9012 // Support: IE 9 - 11 only
9013 // IE throws on parseFromString with invalid input.
9014 try {
9015     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9016 } catch ( e ) {}
9017
9018 // Support: IE 9 - 11 only
9019 // IE throws on parseFromString with invalid input.
9020 try {
9021     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9022 } catch ( e ) {}
9023
9024 // Support: IE 9 - 11 only
9025 // IE throws on parseFromString with invalid input.
9026 try {
9027     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9028 } catch ( e ) {}
9029
9030 // Support: IE 9 - 11 only
9031 // IE throws on parseFromString with invalid input.
9032 try {
9033     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9034 } catch ( e ) {}
9035
9036 // Support: IE 9 - 11 only
9037 // IE throws on parseFromString with invalid input.
9038 try {
9039     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9040 } catch ( e ) {}
9041
9042 // Support: IE 9 - 11 only
9043 // IE throws on parseFromString with invalid input.
9044 try {
9045     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9046 } catch ( e ) {}
9047
9048 // Support: IE 9 - 11 only
9049 // IE throws on parseFromString with invalid input.
9050 try {
9051     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9052 } catch ( e ) {}
9053
9054 // Support: IE 9 - 11 only
9055 // IE throws on parseFromString with invalid input.
9056 try {
9057     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9058 } catch ( e ) {}
9059
9060 // Support: IE 9 - 11 only
9061 // IE throws on parseFromString with invalid input.
9062 try {
9063     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9064 } catch ( e ) {}
9065
9066 // Support: IE 9 - 11 only
9067 // IE throws on parseFromString with invalid input.
9068 try {
9069     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9070 } catch ( e ) {}
9071
9072 // Support: IE 9 - 11 only
9073 // IE throws on parseFromString with invalid input.
9074 try {
9075     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9076 } catch ( e ) {}
9077
9078 // Support: IE 9 - 11 only
9079 // IE throws on parseFromString with invalid input.
9080 try {
9081     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9082 } catch ( e ) {}
9083
9084 // Support: IE 9 - 11 only
9085 // IE throws on parseFromString with invalid input.
9086 try {
9087     ( new window.DOMParser() ).parseFromString( "data", "text/xml" );
9088 } catch ( e ) {}
9089
9090 // Support: IE 9 - 11 only
9091 // IE
```

```

8887
8888 var
8889 rbracket = /\]$/,
8890 rCRLF = /\r?\n/g,
8891 rsubmitterTypes = /^(?:submit|button|image|reset|file)$/i,
8892 rsubmittable = /^(?:input|select|textarea|keygen)/i;
8893
8894 function buildParams( prefix, obj, traditional, add ) {
8895     var name;
8896
8897     if ( Array.isArray( obj ) ) {
8898
8899         // Serialize array item.
8900         jQuery.each( obj, function( i, v ) {
8901             if ( traditional || rbracket.test( prefix ) ) {
8902
8903                 // Treat each array item as a scalar.
8904                 add( prefix, v );
8905             } else {
8906
8907                 // Item is non-scalar (array or object), encode its numeric index.
8908                 buildParams(
8909                     prefix + "[" + ( typeof v === "object" && v != null ? i : "" ) + "]",
8910                     v,
8911                     traditional,
8912                     add
8913                 );
8914             }
8915         } );
8916     } else if ( !traditional && toType( obj ) === "object" ) {
8917
8918         // Serialize object item.
8919         for ( name in obj ) {
8920             buildParams( prefix + "[" + name + "]", obj[ name ], traditional, add );
8921         }
8922     } else {
8923
8924         // Serialize scalar item.
8925         add( prefix, obj );
8926     }
8927 }
8928
8929 // Serialize an array of form elements or a set of
8930 // key/values into a query string
8931 jQuery.param = function( a, traditional ) {
8932     var prefix;
8933     s = [];
8934     add = function( key, valueOrFunction ) {
8935
8936         // If value is a function, invoke it and use its return value
8937         var value =isFunction( valueOrFunction ) ?
8938             valueOrFunction() :
8939             valueOrFunction;
8940
8941         s[ s.length ] = encodeURIComponent( key ) + "=" +
8942             encodeURIComponent( value == null ? "" : value );
8943     };
8944
8945     if ( a == null ) {
8946         return "";
8947     }
8948
8949     // If an array was passed in, assume that it is an array of form elements.
8950     if ( Array.isArray( a ) || ( a.jquery && !jQuery.isPlainObject( a ) ) ) {
8951
8952         // Serialize the form elements
8953         jQuery.each( a, function() {
8954             add( this.name, this.value );
8955         } );
8956     } else {
8957
8958         // If traditional, encode the "old" way (the way 1.3.2 or older
8959         // did it), otherwise encode params recursively.
8960         for ( prefix in a ) {
8961             buildParams( prefix, a[ prefix ], traditional, add );
8962         }
8963     }
8964
8965     // Return the resulting serialization
8966     return s.join( "&" );
8967 }
8968
8969 jQuery.fn.extend( {
8970     serialize: function() {
8971         return jQuery.param( this.serializeArray() );
8972     },
8973     serializeArray: function() {
8974         return this.map( function() {
8975
8976             // Can add propHook for "elements" to filter or add form elements
8977             var elements = jQuery.prop( this, "elements" );
8978             return elements ? jQuery.makeArray( elements ) : this;
8979         } ).filter( function() {
8980             var type = this.type;
8981
8982             // Use .is( ":disabled" ) so that fieldset[disabled] works
8983             return this.name && !jQuery( this ).is( ":disabled" ) &&
8984                 rsubmittable.test( this.nodeName ) && !rsubmitterTypes.test( type ) &&
8985                 ( this.checked || !rcheckableType.test( type ) );
8986         } ).map( function( i, elem ) {
8987             var val = jQuery( this ).val();
8988
8989             if ( val == null ) {
8990                 return null;
8991             }
8992
8993             if ( Array.isArray( val ) ) {
8994                 return jQuery.map( val, function( val ) {
8995                     return { name: elem.name, value: val.replace( rCRLF, "\r\n" ) };
8996                 } );
8997             }
8998         } );
8999     }
9000 } );
9001

```

```

9002
9003     return { name: elem.name, value: val.replace( rCRLF, "\r\n" ) };
9004   }, get();
9005 }
9006 } );
9007
9008
9009 var
9010 r20 = /%20/g,
9011 rhash = /#.*$/i,
9012 rantiCache = /([?&])_=[^&]*/,
9013 rheaders = /^(.*?):\s([\t]*([^\r\n]*)*)$/mg,
9014
9015 // #7653, #8125, #8152: local protocol detection
9016 rlocalProtocol = /^(?:about|app|app-storage|[-extension]|file|res|widget):$/,
9017 rnoContent = /^(?:GET|HEAD)$/,
9018 rprotocol = /^V/,
9019
9020 /* Prefilters
9021  * 1) They are useful to introduce custom dataTypes (see ajax/jsonp.js for an example)
9022  * 2) These are called:
9023  *    - BEFORE asking for a transport
9024  *    - AFTER param serialization (s.data is a string if s.processData is true)
9025  * 3) key is the dataType
9026  * 4) the catchall symbol "*" can be used
9027  * 5) execution will start with transport dataType and THEN continue down to "*" if needed
9028  */
9029 preFilters = {},
9030
9031 /* Transports bindings
9032  * 1) key is the dataType
9033  * 2) the catchall symbol "*" can be used
9034  * 3) selection will start with transport dataType and THEN go to "*" if needed
9035  */
9036 transports = {},
9037
9038 // Avoid comment-prolog char sequence (#10098); must appease lint and evade compression
9039 allTypes = "*/" concat( "/*" ),
9040
9041 // Anchor tag for parsing the document origin
9042 originAnchor = document.createElement( "a" );
9043
9044 originAnchor.href = location.href,
9045
9046 // Base "constructor" for jQuery.ajaxPrefilter and jQuery.ajaxTransport
9047 function addToPrefiltersOrTransports( structure ) {
9048
9049   // dataTypeExpression is optional and defaults to "*"
9050   return function( dataTypeExpression, func ) {
9051
9052     if ( typeof dataTypeExpression !== "string" ) {
9053       func = dataTypeExpression;
9054       dataTypeExpression = "*";
9055     }
9056
9057     var dataType,
9058         i = 0,
9059         dataTypes = dataTypeExpression.toLowerCase().match( rnothtmlwhite ) || [];
9060
9061     if ( isFunction( func ) ) {
9062
9063       // For each dataType in the dataTypeExpression
9064       while ( ( dataType = dataTypes[ i++ ] ) ) {
9065
9066         // Prepend if requested
9067         if ( dataType[ 0 ] === "+" ) {
9068           dataType = dataType.slice( 1 ) || "*";
9069           ( structure[ dataType ] = structure[ dataType ] || [] ).unshift( func );
9070
9071         // Otherwise append
9072         } else {
9073           ( structure[ dataType ] = structure[ dataType ] || [] ).push( func );
9074         }
9075       }
9076     }
9077   };
9078 }
9079
9080 // Base inspection function for prefilters and transports
9081 function inspectPrefiltersOrTransports( structure, options, originalOptions, jqXHR ) {
9082
9083   var inspected = {},
9084       seekingTransport = ( structure === transports );
9085
9086   function inspect( dataType ) {
9087     var selected,
9088         inspected[ dataType ] = true;
9089     jQuery.each( structure[ dataType ] || [], function( _, prefilterOrFactory ) {
9090       var dataTypeOrTransport = prefilterOrFactory( options, originalOptions, jqXHR );
9091       if ( ( typeof dataTypeOrTransport === "string" &&
9092         !seekingTransport && !inspected[ dataTypeOrTransport ] ) ) {
9093
9094         options.dataTypes.unshift( dataTypeOrTransport );
9095         inspect( dataTypeOrTransport );
9096         return false;
9097       } else if ( seekingTransport ) {
9098         return !( selected = dataTypeOrTransport );
9099       }
9100     });
9101     return selected;
9102   }
9103
9104   return inspect( options.dataTypes[ 0 ] ) || !inspected[ "*" ] && inspect( "*" );
9105 }
9106
9107 // A special extend for ajax options
9108 // that takes "flat" options (not to be deep extended)
9109 // Fixes #9887
9110 function ajaxExtend( target, src ) {
9111   var key, deep,
9112       flatOptions = jQuery.ajaxSettings.flatOptions || {};
9113
9114   for ( key in src ) {
9115     if ( src[ key ] !== undefined ) {
9116       ( flatOptions[ key ] ? target : ( deep || ( deep = {} ) ) )[ key ] = src[ key ];
9117     }
9118   }
9119 }

```

```

9117     }
9118   }
9119   if ( deep ) {
9120     jQuery.extend( true, target, deep );
9121   }
9122 }
9123 return target;
9124 }
9125
9126 /* Handles responses to an ajax request:
9127  * - finds the right dataType (mediates between content-type and expected dataType)
9128  * - returns the corresponding response
9129  */
9130 function ajaxHandleResponses( s, jqXHR, responses ) {
9131
9132   var ct, type, finalDataType, firstDataType,
9133       contents = s.contents,
9134       dataTypes = s.dataTypes;
9135
9136   // Remove auto dataType and get content-type in the process
9137   while ( dataTypes[ 0 ] === "*" ) {
9138     dataTypes.shift();
9139     if ( ct === undefined ) {
9140       ct = s.mimeType || jqXHR.getResponseHeader( "Content-Type" );
9141     }
9142   }
9143
9144   // Check if we're dealing with a known content-type
9145   if ( ct ) {
9146     for ( type in contents ) {
9147       if ( contents[ type ] && contents[ type ].test( ct ) ) {
9148         dataTypes.unshift( type );
9149         break;
9150       }
9151     }
9152   }
9153
9154   // Check to see if we have a response for the expected dataType
9155   if ( dataTypes[ 0 ] in responses ) {
9156     finalDataType = dataTypes[ 0 ];
9157   } else {
9158     // Try convertible dataTypes
9159     for ( type in responses ) {
9160       if ( !dataTypes[ 0 ] || s.converters[ type + " " + dataTypes[ 0 ] ] ) {
9161         finalDataType = type;
9162         break;
9163       }
9164     }
9165     if ( !firstDataType ) {
9166       firstDataType = type;
9167     }
9168   }
9169
9170   // Or just use first one
9171   finalDataType = finalDataType || firstDataType;
9172 }
9173
9174 // If we found a dataType
9175 // We add the dataType to the list if needed
9176 // and return the corresponding response
9177 if ( finalDataType ) {
9178   if ( finalDataType !== dataTypes[ 0 ] ) {
9179     dataTypes.unshift( finalDataType );
9180   }
9181   return responses[ finalDataType ];
9182 }
9183 }
9184
9185 /* Chain conversions given the request and the original response
9186  * Also sets the responseXXX fields on the jqXHR instance
9187  */
9188 function ajaxConvert( s, response, jqXHR, isSuccess ) {
9189   var conv2, current, conv, tmp, prev,
9190       converters = {},
9191
9192   // Work with a copy of dataTypes in case we need to modify it for conversion
9193   dataTypes = s.dataTypes.slice();
9194
9195   // Create converters map with lowercased keys
9196   if ( dataTypes[ 1 ] ) {
9197     for ( conv in s.converters ) {
9198       converters[ conv.toLowerCase() ] = s.converters[ conv ];
9199     }
9200   }
9201
9202   current = dataTypes.shift();
9203
9204   // Convert to each sequential dataType
9205   while ( current ) {
9206
9207     if ( s.responseFields[ current ] ) {
9208       jqXHR[ s.responseFields[ current ] ] = response;
9209     }
9210
9211     // Apply the dataTypeFilter if provided
9212     if ( !prev && isSuccess && s.dataFilter ) {
9213       response = s.dataFilter( response, s.dataType );
9214     }
9215
9216     prev = current;
9217     current = dataTypes.shift();
9218
9219     if ( !current ) {
9220
9221       // There's only work to do if current dataType is non-auto
9222       if ( current === "*" ) {
9223
9224         current = prev;
9225
9226         // Convert response if prev dataType is non-auto and differs from current
9227       } else if ( !prev || prev !== "*" && prev !== current ) {
9228
9229         // Seek a direct converter
9230         conv = converters[ prev + " " + current ] || converters[ "*" + current ];
9231

```

```

9232 // If none found, seek a pair
9233 if ( !conv ) {
9234   for ( conv2 in converters ) {
9235
9236     // If conv2 outputs current
9237     tmp = conv2.split( " " );
9238     if ( tmp[ 1 ] === current ) {
9239
9240       // If prev can be converted to accepted input
9241       conv = converters[ prev + " " + tmp[ 0 ] ] ||
9242         converters[ "*" + tmp[ 0 ] ];
9243       if ( conv ) {
9244
9245         // Condense equivalence converters
9246         if ( conv === true ) {
9247           conv = converters[ conv2 ];
9248
9249           // Otherwise, insert the intermediate dataType
9250           } else if ( converters[ conv2 ] !== true ) {
9251             current = tmp[ 0 ];
9252             dataTypes.unshift( tmp[ 1 ] );
9253           }
9254           break;
9255         }
9256       }
9257     }
9258   }
9259
9260   // Apply converter (if not an equivalence)
9261   if ( conv !== true ) {
9262
9263     // Unless errors are allowed to bubble, catch and return them
9264     if ( conv && s.throws ) {
9265       response = conv( response );
9266     } else {
9267       try {
9268         response = conv( response );
9269       } catch ( e ) {
9270         return {
9271           state: "parsererror",
9272           error: conv ? e : "No conversion from " + prev + " to " + current
9273         };
9274       }
9275     }
9276   }
9277 }
9278 }
9279 }
9280
9281 return { state: "success", data: response };
9282 }
9283
9284 jQuery.extend( {
9285
9286   // Counter for holding the number of active queries
9287   active: 0,
9288
9289   // Last-Modified header cache for next request
9290   lastModified: {},
9291   etag: {},
9292
9293   ajaxSettings: {
9294     url: location.href,
9295     type: "GET",
9296     isLocal: rlocalProtocol.test( location.protocol ),
9297     global: true,
9298     processData: true,
9299     async: true,
9300     contentType: "application/x-www-form-urlencoded; charset=UTF-8",
9301
9302     /*
9303     timeout: 0,
9304     data: null,
9305     dataType: null,
9306     username: null,
9307     password: null,
9308     cache: null,
9309     throws: false,
9310     traditional: false,
9311     headers: {},
9312     */
9313
9314     accepts: {
9315       "*": allTypes,
9316       text: "text/plain",
9317       html: "text/html",
9318       xml: "application/xml, text/xml",
9319       json: "application/json, text/javascript"
9320     },
9321
9322     contents: {
9323       xml: /bxml\b/,
9324       html: /bhtml/,
9325       json: /bson\b/
9326     },
9327
9328     responseFields: {
9329       xml: "responseXML",
9330       text: "responseText",
9331       json: "responseJSON"
9332     },
9333
9334     // Data converters
9335     // Keys separate source (or catchall "*") and destination types with a single space
9336     converters: {
9337
9338       // Convert anything to text
9339       "* text": String,
9340
9341       // Text to html (true = no transformation)
9342       "text html": true,
9343
9344       // Evaluate text as a json expression
9345       "text json": JSON.parse,
9346

```



```

9347 // Parse text as xml
9348 "text xml": jQuery.parseXML
9349 },
9350
9351 // For options that shouldn't be deep extended:
9352 // you can add your own custom options here if
9353 // and when you create one that shouldn't be
9354 // deep extended (see ajaxExtend)
9355 flatOptions: {
9356   url: true,
9357   context: true
9358 },
9359 },
9360
9361 // Creates a full fledged settings object into target
9362 // with both ajaxSettings and settings fields.
9363 // If target is omitted, writes into ajaxSettings.
9364 ajaxSetup: function( target, settings ) {
9365   return settings ?
9366     // Building a settings object
9367     ajaxExtend( ajaxExtend( target, jQuery.ajaxSettings ), settings ) :
9368     // Extending ajaxSettings
9369     ajaxExtend( jQuery.ajaxSettings, target );
9370 },
9371
9372 ajaxPrefilter: addToPrefiltersOrTransports( prefilters ),
9373 ajaxTransport: addToPrefiltersOrTransports( transports ),
9374
9375 // Main method
9376 ajax: function( url, options ) {
9377   // If url is an object, simulate pre-1.5 signature
9378   if ( typeof url === "object" ) {
9379     options = url;
9380     url = undefined;
9381   }
9382   // Force options to be an object
9383   options = options || {};
9384
9385   // URL without anti-cache param
9386   cacheURL =
9387     // Response headers
9388     responseHeadersString,
9389     responseHeaders,
9390     // timeout handle
9391     timeoutTimer,
9392     // Url cleanup var
9393     urlAnchor,
9394     // Request state (becomes false upon send and true upon completion)
9395     completed,
9396     // To know if global events are to be dispatched
9397     fireGlobals,
9398     // Loop variable
9399     i,
9400     // uncached part of the url
9401     uncached,
9402     // Create the final options object
9403     s = jQuery.ajaxSetup( {}, options ),
9404     // Callbacks context
9405     callbackContext = s.context || s,
9406     // Context for global events is callbackContext if it is a DOM node or jQuery collection
9407     globalEventContext = s.context &&
9408       ( callbackContext.nodeType || callbackContext.jquery ) ?
9409         jQuery( callbackContext ) :
9410         jQuery.event,
9411     // Deferreds
9412     deferred = jQuery.Deferred(),
9413     completeDeferred = jQuery.Callbacks( "once memory" ),
9414     // Status-dependent callbacks
9415     statusCode = s.statusCode || {},
9416     // Headers (they are sent all at once)
9417     requestHeaders = {},
9418     requestHeadersNames = {},
9419     // Default abort message
9420     strAbort = "canceled",
9421     // Fake xhr
9422     jqXHR = {
9423       readyState: 0,
9424
9425       // Builds headers hashtable if needed
9426       getResponseHeader: function( key ) {
9427         var match;
9428         if ( !completed ) {
9429           if ( !responseHeaders ) {
9430             responseHeaders = {};
9431             while ( ( match = rheaders.exec( responseHeadersString ) ) ) {
9432               responseHeaders[ match[ 1 ].toLowerCase() + " " ] =
9433                 ( responseHeaders[ match[ 1 ].toLowerCase() + " " ] || [] )
9434                   .concat( match[ 2 ] );
9435             }
9436           }
9437           match = responseHeaders[ key.toLowerCase() + " " ];
9438         }
9439         return match == null ? null : match.join( ", " );
9440       },
9441     },

```

```

9462
9463 // Raw string
9464 getAllResponseHeaders: function() {
9465     return completed ? responseHeadersString : null;
9466 },
9467
9468 // Caches the header
9469 setRequestHeader: function( name, value ) {
9470     if ( completed == null ) {
9471         name = requestHeadersNames[ name.toLowerCase() ] =
9472             requestHeadersNames[ name.toLowerCase() ] || name;
9473         requestHeaders[ name ] = value;
9474     }
9475     return this;
9476 },
9477
9478 // Overrides response content-type header
9479 overrideMimeType: function( type ) {
9480     if ( completed == null ) {
9481         s.mimeType = type;
9482     }
9483     return this;
9484 },
9485
9486 // Status-dependent callbacks
9487 statusCode: function( map ) {
9488     var code;
9489     if ( map ) {
9490         if ( completed ) {
9491
9492             // Execute the appropriate callbacks
9493             jqXHR.always( map[ jqXHR.status ] );
9494         } else {
9495
9496             // Lazy-add the new callbacks in a way that preserves old ones
9497             for ( code in map ) {
9498                 statusCode[ code ] = [ statusCode[ code ], map[ code ] ];
9499             }
9500         }
9501     }
9502     return this;
9503 },
9504
9505 // Cancel the request
9506 abort: function( textStatus ) {
9507     var finalText = textStatus || "Abort";
9508     if ( transport ) {
9509         transport.abort( finalText );
9510     }
9511     done( 0, finalText );
9512     return this;
9513 }
9514 };
9515
9516 // Attach deferreds
9517 deferred.promise( jqXHR );
9518
9519 // Add protocol if not provided (prefilters might expect it)
9520 // Handle falsy url in the settings object (#10093: consistency with old signature)
9521 // We also use the url parameter if available
9522 s.url = ( ( url || s.url || location.href ) + "" )
9523     .replace( rprotocol, location.protocol + "//" );
9524
9525 // Alias method option to type as per ticket #12004
9526 s.type = options.method || options.type || s.method || s.type;
9527
9528 // Extract dataType list
9529 s.dataTypes = ( s.dataType || "*" ).toLowerCase().match( rnothtmlwhite ) || [ "" ];
9530
9531 // A cross-domain request is in order when the origin doesn't match the current origin.
9532 if ( s.crossDomain == null ) {
9533     urlAnchor = document.createElement( "a" );
9534
9535     // Support: IE <= 8 - 11, Edge 12 - 15
9536     // IE throws exception on accessing the href property if url is malformed,
9537     // e.g. http://example.com:80x/
9538     try {
9539         urlAnchor.href = s.url;
9540
9541         // Support: IE <= 8 - 11 only
9542         // Anchor's host property isn't correctly set when s.url is relative
9543         urlAnchor.href = urlAnchor.href;
9544         s.crossDomain = originAnchor.protocol + "://" + originAnchor.host !==
9545             urlAnchor.protocol + "://" + urlAnchor.host;
9546     } catch ( e ) {
9547         // If there is an error parsing the URL, assume it is crossDomain,
9548         // it can be rejected by the transport if it is invalid
9549         s.crossDomain = true;
9550     }
9551 }
9552
9553 // Convert data if not already a string
9554 if ( s.data && s.processData && typeof s.data !== "string" ) {
9555     s.data = jQuery.param( s.data, s.traditional );
9556 }
9557
9558 // Apply prefilters
9559 inspectPrefiltersOrTransports( prefilters, s, options, jqXHR );
9560
9561 // If request was aborted inside a prefilter, stop there
9562 if ( completed ) {
9563     return jqXHR;
9564 }
9565
9566 // We can fire global events as of now if asked to
9567 // Don't fire events if jQuery.event is undefined in an AMD-usage scenario (#15118)
9568 fireGlobals = jQuery.event && s.global;
9569
9570 // Watch for a new set of requests
9571 if ( fireGlobals && jQuery.active++ === 0 ) {
9572     jQuery.event.trigger( "ajaxStart" );
9573 }
9574
9575 // Uppercase the type

```

```

9577 s.type = s.type.toUpperCase();
9578
9579 // Determine if request has content
9580 s.hasContent = !noContent.test( s.type );
9581
9582 // Save the URL in case we're toying with the If-Modified-Since
9583 // and/or If-None-Match header later on
9584 // Remove hash to simplify url manipulation
9585 cacheURL = s.url.replace( rhash, "" );
9586
9587 // More options handling for requests with no content
9588 if ( ! s.hasContent ) {
9589
9590     // Remember the hash so we can put it back
9591     uncached = s.url.slice( cacheURL.length );
9592
9593     // If data is available and should be processed, append data to url
9594     if ( s.data && ( s.processData || typeof s.data === "string" ) ) {
9595         cacheURL += ( rquery.test( cacheURL ) ? "&" : "?" ) + s.data;
9596
9597         // #9682: remove data so that it's not used in an eventual retry
9598         delete s.data;
9599     }
9600
9601     // Add or update anti-cache param if needed
9602     if ( ! s.cache === false ) {
9603         cacheURL = cacheURL.replace( rantiCache, "$1" );
9604         uncached = ( rquery.test( cacheURL ) ? "&" : "?" ) + "_" + ( nonce.guid++ ) +
9605             uncached;
9606     }
9607
9608     // Put hash and anti-cache on the URL that will be requested (gh-1732)
9609     s.url = cacheURL + uncached;
9610
9611     // Change '%20' to '+' if this is encoded form body content (gh-2658)
9612 } else if ( s.data && s.processData &&
9613     ( s.contentType || "" ).indexOf( "application/x-www-form-urlencoded" ) === 0 ) {
9614     s.data = s.data.replace( r20, "+" );
9615 }
9616
9617 // Set the If-Modified-Since and/or If-None-Match header, if in ifModified mode.
9618 if ( s.ifModified ) {
9619     if ( jQuery.lastModified[ cacheURL ] ) {
9620         jqXHR.setRequestHeader( "If-Modified-Since", jQuery.lastModified[ cacheURL ] );
9621     }
9622     if ( jQuery.etag[ cacheURL ] ) {
9623         jqXHR.setRequestHeader( "If-None-Match", jQuery.etag[ cacheURL ] );
9624     }
9625 }
9626
9627 // Set the correct header, if data is being sent
9628 if ( s.data && s.hasContent && s.contentType !== false || options.contentType ) {
9629     jqXHR.setRequestHeader( "Content-Type", s.contentType );
9630 }
9631
9632 // Set the Accepts header for the server, depending on the dataType
9633 jqXHR.setRequestHeader(
9634     "Accept",
9635     s.dataTypes[ 0 ] && s.accepts[ s.dataTypes[ 0 ] ] ?
9636         s.accepts[ s.dataTypes[ 0 ] ] +
9637         ( s.dataTypes[ 0 ] !== "*" ? ", " + allTypes + "; q=0.01" : "" ) :
9638         s.accepts[ "*" ]
9639 );
9640
9641 // Check for headers option
9642 for ( i in s.headers ) {
9643     jqXHR.setRequestHeader( i, s.headers[ i ] );
9644 }
9645
9646 // Allow custom headers/mimetypes and early abort
9647 if ( s.beforeSend &&
9648     ( s.beforeSend.call( callbackContext, jqXHR, s ) === false || completed ) ) {
9649
9650     // Abort if not done already and return
9651     return jqXHR.abort();
9652 }
9653
9654 // Aborting is no longer a cancellation
9655 strAbort = "abort";
9656
9657 // Install callbacks on deferreds
9658 completeDeferred.add( s.complete );
9659 jqXHR.done( s.success );
9660 jqXHR.fail( s.error );
9661
9662 // Get transport
9663 transport = inspectPrefiltersOrTransports( transports, s, options, jqXHR );
9664
9665 // If no transport, we auto-abort
9666 if ( ! transport ) {
9667     done( -1, "No Transport" );
9668 } else {
9669     jqXHR.readyState = 1;
9670
9671     // Send global event
9672     if ( fireGlobals ) {
9673         globalEventContext.trigger( "ajaxSend", [ jqXHR, s ] );
9674     }
9675
9676     // If request was aborted inside ajaxSend, stop there
9677     if ( completed ) {
9678         return jqXHR;
9679     }
9680
9681     // Timeout
9682     if ( s.async && s.timeout > 0 ) {
9683         setTimeout( function() {
9684             jqXHR.abort( "timeout" );
9685         }, s.timeout );
9686     }
9687
9688     try {
9689         completed = false;
9690         transport.send( requestHeaders, done );
9691     } catch ( e ) {

```

```

9692
9693 // Rethrow post-completion exceptions
9694 if ( completed ) {
9695     throw e;
9696 }
9697
9698 // Propagate others as results
9699 done( -1, e );
9700 }
9701 }
9702
9703 // Callback for when everything is done
9704 function done( status, nativeStatusText, responses, headers ) {
9705     var isSuccess, success, error, response, modified;
9706     statusText = nativeStatusText;
9707
9708     // Ignore repeat invocations
9709     if ( completed ) {
9710         return;
9711     }
9712
9713     completed = true;
9714
9715     // Clear timeout if it exists
9716     if ( timeoutTimer ) {
9717         window.clearTimeout( timeoutTimer );
9718     }
9719
9720     // Dereference transport for early garbage collection
9721     // (no matter how long the jqXHR object will be used)
9722     transport = undefined;
9723
9724     // Cache response headers
9725     responseHeadersString = headers || "";
9726
9727     // Set readyState
9728     jqXHR.readyState = status >= 0 ? 4 : 0;
9729
9730     // Determine if successful
9731     isSuccess = status >= 200 && status < 300 || status === 304;
9732
9733     // Get response data
9734     if ( responses ) {
9735         response = ajaxHandleResponses( s, jqXHR, responses );
9736     }
9737
9738     // Use a noop converter for missing script but not if jsonp
9739     if ( !isSuccess &&
9740         jQuery.inArray( "script", s.dataTypes ) > -1 &&
9741         jQuery.inArray( "json", s.dataTypes ) < 0 ) {
9742         s.converters[ "text script" ] = function() {};
9743     }
9744
9745     // Convert no matter what (that way responseXXX fields are always set)
9746     response = ajaxConvert( s, response, jqXHR, isSuccess );
9747
9748     // If successful, handle type chaining
9749     if ( isSuccess ) {
9750
9751         // Set the If-Modified-Since and/or If-None-Match header, if in ifModified mode.
9752         if ( s.ifModified ) {
9753             modified = jqXHR.getResponseHeader( "Last-Modified" );
9754             if ( modified ) {
9755                 jQuery.lastModified[ cacheURL ] = modified;
9756             }
9757             modified = jqXHR.getResponseHeader( "etag" );
9758             if ( modified ) {
9759                 jQuery.etag[ cacheURL ] = modified;
9760             }
9761         }
9762
9763         // if no content
9764         if ( status === 204 || s.type === "HEAD" ) {
9765             statusText = "nocontent";
9766         }
9767         // if not modified
9768     } else if ( status === 304 ) {
9769         statusText = "notmodified";
9770
9771         // If we have data, let's convert it
9772     } else {
9773         statusText = response.state;
9774         success = response.data;
9775         error = response.error;
9776         isSuccess = !error;
9777     }
9778     } else {
9779
9780         // Extract error from statusText and normalize for non-aborts
9781         error = statusText;
9782         if ( status || !statusText ) {
9783             statusText = "error";
9784             if ( status < 0 ) {
9785                 status = 0;
9786             }
9787         }
9788     }
9789
9790     // Set data for the fake xhr object
9791     jqXHR.status = status;
9792     jqXHR.statusText = ( nativeStatusText || statusText ) + "";
9793
9794     // Success/Error
9795     if ( isSuccess ) {
9796         deferred.resolveWith( callbackContext, [ success, statusText, jqXHR ] );
9797     } else {
9798         deferred.rejectWith( callbackContext, [ jqXHR, statusText, error ] );
9799     }
9800
9801     // Status-dependent callbacks
9802     jqXHR.statusCode( statusCode );
9803     statusCode = undefined;
9804
9805     if ( fireGlobals ) {
9806         globalEventContext.trigger( isSuccess ? "ajaxSuccess" : "ajaxError",

```

```

9807     { jqXHR, s: isSuccess ? success : error } });
9808 }
9809
9810 // Complete
9811 completeDeferred.fireWith( callbackContext, [ jqXHR, textStatus ] );
9812
9813 if ( fireGlobals ) {
9814     globalEventContext.trigger( "ajaxComplete", [ jqXHR, s ] );
9815 }
9816 // Handle the global AJAX counter
9817 if ( ! --jQuery.active ) {
9818     jQuery.event.trigger( "ajaxStop" );
9819 }
9820 }
9821 }
9822
9823 return jqXHR;
9824 },
9825
9826 getJSON: function( url, data, callback ) {
9827     return jQuery.get( url, data, callback, "json" );
9828 },
9829
9830 getScript: function( url, callback ) {
9831     return jQuery.get( url, undefined, callback, "script" );
9832 }
9833 };
9834
9835 jQuery.each( [ "get", "post" ], function( _i, method ) {
9836     jQuery[ method ] = function( url, data, callback, type ) {
9837         // Shift arguments if data argument was omitted
9838         if ( !isFunction( data ) ) {
9839             type = type || callback;
9840             callback = data;
9841             data = undefined;
9842         }
9843     };
9844
9845     // The url can be an options object (which then must have .url)
9846     return jQuery.ajax( jQuery.extend( {
9847         url: url,
9848         type: method,
9849         dataType: type,
9850         data: data,
9851         success: callback
9852     }, jQuery.isPlainObject( url ) && url ) );
9853 };
9854 });
9855
9856 jQuery.ajaxPrefilter( function( s ) {
9857     var i;
9858     for ( i in s.headers ) {
9859         if ( i.toLowerCase() === "content-type" ) {
9860             s.contentType = s.headers[ i ] || "";
9861         }
9862     }
9863 });
9864
9865
9866 jQuery_evalUrl = function( url, options, doc ) {
9867     return jQuery.ajax( {
9868         url: url,
9869
9870         // Make this explicit, since user can override this through ajaxSetup (#11264)
9871         type: "GET",
9872         dataType: "script",
9873         cache: true,
9874         async: false,
9875         global: false,
9876
9877         // Only evaluate the response if it is successful (gh-4126)
9878         // dataFilter is not invoked for failure responses, so using it instead
9879         // of the default converter is kludgy but it works.
9880         converters: {
9881             "text script": function() {}
9882         },
9883         dataFilter: function( response ) {
9884             jQuery.globalEval( response, options, doc );
9885         }
9886     } );
9887 };
9888
9889
9890 jQuery.fn.extend( {
9891     wrapAll: function( html ) {
9892         var wrap;
9893
9894         if ( this[ 0 ] ) {
9895             if ( isFunction( html ) ) {
9896                 html = html.call( this[ 0 ] );
9897             }
9898
9899             // The elements to wrap the target around
9900             wrap = jQuery( html, this[ 0 ].ownerDocument ).eq( 0 ).clone( true );
9901
9902             if ( this[ 0 ].parentNode ) {
9903                 wrap.insertBefore( this[ 0 ] );
9904             }
9905
9906             wrap.map( function() {
9907                 var elem = this;
9908
9909                 while ( elem.firstChild ) {
9910                     elem = elem.firstChild;
9911                 }
9912
9913                 return elem;
9914             } ).append( this );
9915
9916             return this;
9917         },
9918
9919     wrapInner: function( html ) {
9920         if ( isFunction( html ) ) {

```

```

9922     return this.each( function( i ) {
9923         jQuery( this ).wrapInner( html.call( this, i ) );
9924     } );
9925 }
9926
9927 return this.each( function() {
9928     var self = jQuery( this );
9929     contents = self.contents();
9930
9931     if ( contents.length ) {
9932         contents.wrapAll( html );
9933     } else {
9934         self.append( html );
9935     }
9936 } );
9937 } );
9938 },
9939
9940 wrap: function( html ) {
9941     var htmlsFunction = isFunction( html );
9942
9943     return this.each( function( i ) {
9944         jQuery( this ).wrapAll( htmlsFunction ? html.call( this, i ) : html );
9945     } );
9946 },
9947
9948 unwrap: function( selector ) {
9949     this.parent( selector ).not( "body" ).each( function() {
9950         jQuery( this ).replaceWith( this.childNodes );
9951     } );
9952     return this;
9953 }
9954 } );
9955
9956
9957 jQuery.expr.pseudos.hidden = function( elem ) {
9958     return !jQuery.expr.pseudos.visible( elem );
9959 };
9960
9961 jQuery.expr.pseudos.visible = function( elem ) {
9962     return !!( elem.offsetWidth || elem.offsetHeight || elem.getClientRects().length );
9963 };
9964
9965
9966
9967 jQuery.ajaxSettings.xhr = function() {
9968     try {
9969         return new window.XMLHttpRequest();
9970     } catch ( e ) {}
9971 };
9972
9973 var xhrSuccessStatus = {
9974     // File protocol always yields status code 0, assume 200
9975     0: 200,
9976
9977     // Support: IE <=9 only
9978     // #1450: sometimes IE returns 1223 when it should be 204
9979     1223: 204
9980 },
9981
9982 xhrSupported = jQuery.ajaxSettings.xhr();
9983
9984 support.cors = !!xhrSupported && ( "withCredentials" in xhrSupported );
9985 support.ajax = xhrSupported = !!xhrSupported;
9986
9987 jQuery.ajaxTransport( function( options ) {
9988     var callback, errorCallback;
9989
9990     // Cross domain only allowed if supported through XMLHttpRequest
9991     if ( support.cors || xhrSupported && !options.crossDomain ) {
9992         return {
9993             send: function( headers, complete ) {
9994                 var i;
9995                 xhr = options.xhr();
9996
9997                 xhr.open(
9998                     options.type,
9999                     options.url,
10000                     options.async,
10001                     options.username,
10002                     options.password
10003                 );
10004
10005                 // Apply custom fields if provided
10006                 if ( options.xhrFields ) {
10007                     for ( i in options.xhrFields ) {
10008                         xhr[ i ] = options.xhrFields[ i ];
10009                     }
10010                 }
10011
10012                 // Override mime type if needed
10013                 if ( options.mimeType && xhr.overrideMimeType ) {
10014                     xhr.overrideMimeType( options.mimeType );
10015                 }
10016
10017                 // X-Requested-With header
10018                 // For cross-domain requests, seeing as conditions for a preflight are
10019                 // akin to a jigsaw puzzle, we simply never set it to be sure.
10020                 // (it can always be set on a per-request basis or even using ajaxSetup)
10021                 // For same-domain requests, won't change header if already provided.
10022                 if ( !options.crossDomain && !headers[ "X-Requested-With" ] ) {
10023                     headers[ "X-Requested-With" ] = "XMLHttpRequest";
10024                 }
10025
10026                 // Set headers
10027                 for ( i in headers ) {
10028                     xhr.setRequestHeader( i, headers[ i ] );
10029                 }
10030
10031                 // Callback
10032                 callback = function( type ) {
10033                     return function() {
10034                         if ( callback ) {
10035                             callback = errorCallback = xhr.onload =
10036                                 xhr.onerror = xhr.onabort = xhr.ontimeout =

```

```

10037     xhr.onreadystatechange = null;
10038
10039     if ( type === "abort" ) {
10040         xhr.abort();
10041     } else if ( type === "error" ) {
10042
10043         // Support: IE <=9 only
10044         // On a manual native abort, IE9 throws
10045         // errors on any property access that is not readyState
10046         if ( typeof xhr.status !== "number" ) {
10047             complete( 0, "error" );
10048         } else {
10049             complete();
10050
10051             // File: protocol always yields status 0; see #8605, #14207
10052             xhr.status;
10053             xhr.statusText;
10054         };
10055     }
10056 } else {
10057     complete();
10058     xhrSuccessStatus( xhr.status ) || xhr.status;
10059     xhr.statusText;
10060
10061     // Support: IE <=9 only
10062     // IE9 has no XHR2 but throws on binary (trac-11426)
10063     // For XHR2 non-text, let the caller handle it (gh-2498)
10064     ( xhr.responseType || "text" ) !== "text" ||
10065     typeof xhr.responseText !== "string" ?
10066     { binary: xhr.response } :
10067     { text: xhr.responseText },
10068     xhr.getAllResponseHeaders()
10069 );
10070 }
10071 }
10072 };
10073 };
10074
10075 // Listen to events
10076 xhr.onload = callback();
10077 errorCallback = xhr.onerror = xhr.ontimeout = callback( "error" );
10078
10079 // Support: IE 9 only
10080 // Use onreadystatechange to replace onabort
10081 // to handle uncaught aborts
10082 if ( xhr.onabort !== undefined ) {
10083     xhr.onabort = errorCallback;
10084 } else {
10085     xhr.onreadystatechange = function() {
10086
10087         // Check readyState before timeout as it changes
10088         if ( xhr.readyState === 4 ) {
10089
10090             // Allow onerror to be called first,
10091             // but that will not handle a native abort
10092             // Also, save errorCallback to a variable
10093             // as xhr.onerror cannot be accessed
10094             window.setTimeout( function() {
10095                 if ( callback ) {
10096                     errorCallback();
10097                 }
10098             } );
10099         }
10100     };
10101 }
10102
10103 // Create the abort callback
10104 callback = callback( "abort" );
10105
10106 try {
10107
10108     // Do send the request (this may raise an exception)
10109     xhr.send( options.hasContent && options.data || null );
10110 } catch ( e ) {
10111
10112     // #14683: Only rethrow if this hasn't been notified as an error yet
10113     if ( callback ) {
10114         throw e;
10115     }
10116 }
10117 },
10118
10119 abort: function() {
10120     if ( callback ) {
10121         callback();
10122     }
10123 }
10124 };
10125 }
10126 );
10127
10128
10129
10130
10131 // Prevent auto-execution of scripts when no explicit dataType was provided (See gh-2432)
10132 jQuery.ajaxPrefilter( function( s ) {
10133     if ( s.crossDomain ) {
10134         s.contents.script = false;
10135     }
10136 } );
10137
10138 // Install script dataType
10139 jQuery.ajaxSetup( {
10140     accepts: {
10141         script: "text/javascript, application/javascript, " +
10142             "application/ecmascript, application/x-ecmascript"
10143     },
10144     contents: {
10145         script: /\b(?:java|ecma)script\b/
10146     },
10147     converters: {
10148         "text script": function( text ) {
10149             jQuery.globalEval( text );
10150             return text;
10151         }
10152     }

```

```

10152 }
10153 });
10154
10155 // Handle cache's special case and crossDomain
10156 jQuery.ajaxPrefilter( "script", function( s ) {
10157     if ( s.cache === undefined ) {
10158         s.cache = false;
10159     }
10160     if ( s.crossDomain ) {
10161         s.type = "GET";
10162     }
10163 });
10164
10165 // Bind script tag hack transport
10166 jQuery.ajaxTransport( "script", function( s ) {
10167     // This transport only deals with cross domain or forced-by-attrs requests
10168     if ( s.crossDomain || s.scriptAttrs ) {
10169         var script = document.createElement( "script" );
10170         script.async = true;
10171         if ( s.scriptAttrs ) {
10172             jQuery.each( s.scriptAttrs, function( key, value ) {
10173                 script.setAttribute( key, value );
10174             });
10175         }
10176         script.src = s.url;
10177         if ( s.crossDomain ) {
10178             script.type = "text/javascript";
10179         }
10180         if ( s.scriptCharset ) {
10181             script.charset = s.scriptCharset;
10182         }
10183         // Use native DOM manipulation to avoid our domManip AJAX trickery
10184         document.head.appendChild( script );
10185     }
10186     return {
10187         abort: function() {
10188             if ( script ) {
10189                 script.parentNode.removeChild( script );
10190             }
10191         },
10192         done: function( data ) {
10193             if ( script ) {
10194                 script.parentNode.removeChild( script );
10195             }
10196         },
10197         error: function( error ) {
10198             if ( script ) {
10199                 script.parentNode.removeChild( script );
10200             }
10201         }
10202     };
10203 });
10204
10205 // Default jsonp settings
10206 jQuery.ajaxSetup( {
10207     jsonp: "callback",
10208     jsonpCallback: function() {
10209         var callback = jQuery.expando + "_" + ( nonce.guid++ );
10210         this[ callback ] = true;
10211         return callback;
10212     }
10213 });
10214
10215 // Detect, normalize options and install callbacks for jsonp requests
10216 jQuery.ajaxPrefilter( "json jsonp", function( s, originalSettings, jqXHR ) {
10217     var callbackName, overwritten, responseContainer,
10218         jsonProp = s.jsonp !== false && ( rjsonp.test( s.url ) ?
10219             "url" :
10220             typeof s.data === "string" &&
10221                 ( s.contentType || "" )
10222                     .indexOf( "application/x-www-form-urlencoded" ) === 0 &&
10223                     rjsonp.test( s.data ) && "data"
10224         );
10225     // Handle iff the expected data type is "jsonp" or we have a parameter to set
10226     if ( jsonProp || s.dataTypes[ 0 ] === "jsonp" ) {
10227         // Get callback name, remembering preexisting value associated with it
10228         callbackName = s.jsonpCallback = jQuery.isFunction( s.jsonpCallback ) ?
10229             s.jsonpCallback :
10230             s.jsonpCallback;
10231         // Insert callback into url or form data
10232         if ( jsonProp ) {
10233             s[ jsonProp ] = s[ jsonProp ].replace( rjsonp, "$1" + callbackName );
10234         } else if ( s.jsonp !== false ) {
10235             s.url += ( rquery.test( s.url ) ? "&" : "?" ) + s.jsonp + "=" + callbackName;
10236         }
10237         // Use data converter to retrieve json after script execution
10238         s.converters[ "script json" ] = function() {
10239             if ( !responseContainer ) {
10240                 jQuery.error( callbackName + " was not called" );
10241             }
10242             return responseContainer[ 0 ];
10243         };
10244         // Force json dataType
10245         s.dataTypes[ 0 ] = "json";
10246         // Install callback
10247         overwritten = window[ callbackName ];
10248         window[ callbackName ] = function() {
10249             responseContainer = arguments;
10250         };
10251         // Clean-up function (fires after converters)
10252         jqXHR.always( function() {
10253             // If previous value didn't exist - remove it
10254             if ( overwritten === undefined ) {
10255                 jQuery.removeProp( window, callbackName );
10256             }
10257             // Otherwise restore preexisting value
10258             else {
10259                 window[ callbackName ] = overwritten;
10260             }
10261         });
10262     }

```



```

10267
10268 // Save back as free
10269 if ( s[ callbackName ] ) {
10270
10271     // Make sure that re-using the options doesn't screw things around
10272     s.jsonpCallback = originalSettings.jsonpCallback;
10273
10274     // Save the callback name for future use
10275     oldCallbacks.push( callbackName );
10276 }
10277
10278 // Call if it was a function and we have a response
10279 if ( responseContainer &&isFunction( overwritten ) ) {
10280     overwritten( responseContainer, 0 );
10281 }
10282
10283 responseContainer = overwritten = undefined;
10284 } };
10285
10286 // Delegate to script
10287 return "script";
10288 }
10289 } );
10290
10291
10292
10293
10294 // Support: Safari 8 only
10295 // In Safari 8 documents created via document.implementation.createHTMLDocument
10296 // collapse sibling forms: the second one becomes a child of the first one.
10297 // Because of that, this security measure has to be disabled in Safari 8.
10298 // https://bugs.webkit.org/show_bug.cgi?id=137337
10299 support.createHTMLDocument = ( function() {
10300     var body = document.implementation.createHTMLDocument( "" ).body;
10301     body.innerHTML = "<form></form><form></form>";
10302     return body.childNodes.length === 2;
10303 } )();
10304
10305
10306 // Argument "data" should be string of html
10307 // context (optional): If specified, the fragment will be created in this context,
10308 // defaults to document
10309 // keepScripts (optional): If true, will include scripts passed in the html string
10310 jQuery.parseHTML = function( data, context, keepScripts ) {
10311     if ( typeof data !== "string" ) {
10312         return [];
10313     }
10314     if ( typeof context === "boolean" ) {
10315         keepScripts = context;
10316         context = false;
10317     }
10318
10319     var base, parsed, scripts;
10320
10321     if ( !context ) {
10322
10323         // Stop scripts or inline event handlers from being executed immediately
10324         // by using document.implementation
10325         if ( support.createHTMLDocument ) {
10326             context = document.implementation.createHTMLDocument( "" );
10327
10328             // Set the base href for the created document
10329             // so any parsed elements with URLs
10330             // are based on the document's URL (gh-2965)
10331             base = context.createElement( "base" );
10332             base.href = document.location.href;
10333             context.head.appendChild( base );
10334         } else {
10335             context = document;
10336         }
10337     }
10338
10339     parsed = rsingleTag.exec( data );
10340     scripts = !keepScripts && [];
10341
10342     // Single tag
10343     if ( parsed ) {
10344         return [ context.createElement( parsed[ 1 ] ) ];
10345     }
10346
10347     parsed = buildFragment( [ data ], context, scripts );
10348
10349     if ( scripts && scripts.length ) {
10350         jQuery( scripts ).remove();
10351     }
10352
10353     return jQuery.merge( [], parsed.childNodes );
10354 };
10355
10356
10357 /**
10358  * Load a url into a page
10359  */
10360 jQuery.fn.load = function( url, params, callback ) {
10361     var selector, type, response;
10362     self = this;
10363     off = url.indexOf( " " );
10364
10365     if ( off >= -1 ) {
10366         selector = stripAndCollapse( url.slice( off ) );
10367         url = url.slice( 0, off );
10368     }
10369
10370     // If it's a function
10371     if ( isFunction( params ) ) {
10372
10373         // We assume that it's the callback
10374         callback = params;
10375         params = undefined;
10376     }
10377     // Otherwise, build a param string
10378     else if ( params && typeof params === "object" ) {
10379         type = "POST";
10380     }
10381
10382

```

```

10382 // If we have elements to modify, make the request
10383 if ( self.length > 0 ) {
10384     jQuery.ajax( {
10385         url: url,
10386
10387         // If "type" variable is undefined, then "GET" method will be used.
10388         // Make value of this field explicit since
10389         // user can override it through ajaxSetup method
10390         type: type || "GET",
10391         dataType: "html",
10392         data: params
10393     } ).done( function(.responseText) {
10394
10395         // Save response for use in complete callback
10396         response = arguments
10397
10398         self.html( selector ?
10399
10400             // If a selector was specified, locate the right elements in a dummy div
10401             // Exclude scripts to avoid IE 'Permission Denied' errors
10402             jQuery( "<div>" ).append( jQuery.parseHTML( responseText ) ).find( selector ) :
10403
10404             // Otherwise use the full result
10405             responseText );
10406
10407         // If the request succeeds, this function gets "data", "status", "jqXHR"
10408         // but they are ignored because response was set above.
10409         // If it fails, this function gets "jqXHR", "status", "error"
10410     } ).always( callback && function( jqXHR, status ) {
10411         self.each( function() {
10412             callback.apply( this, response || [ jqXHR.responseText, status, jqXHR ] );
10413         } );
10414     } );
10415 }
10416
10417 return this;
10418 };
10419
10420
10421
10422
10423 jQuery.expr.pseudos.animated = function( elem ) {
10424     return jQuery.grep( jQuery.timers, function( fn ) {
10425         return elem === fn.elem;
10426     } ).length
10427 };
10428
10429
10430
10431
10432 jQuery.offset = {
10433     setOffset: function( elem, options, i ) {
10434         var curPosition, curLeft, curCSSTop, curTop, curOffset, curCSSLeft, calculatePosition,
10435             position = jQuery.css( elem, "position" ),
10436             curElem = jQuery( elem ),
10437             props = {};
10438
10439         // Set position first, in-case top/left are set even on static elem
10440         if ( position === "static" ) {
10441             elem.style.position = "relative";
10442         }
10443
10444         curOffset = curElem.offset();
10445         curCSSTop = jQuery.css( elem, "top" );
10446         curCSSLeft = jQuery.css( elem, "left" );
10447         calculatePosition = ( position === "absolute" || position === "fixed" ) &&
10448             ( curCSSTop + curCSSLeft ).indexOf( "auto" ) > -1;
10449
10450         // Need to be able to calculate position if either
10451         // top or left is auto and position is either absolute or fixed
10452         if ( ! calculatePosition ) {
10453             curPosition = curElem.position();
10454             curTop = curPosition.top;
10455             curLeft = curPosition.left;
10456         }
10457         else {
10458             curTop = parseFloat( curCSSTop ) || 0;
10459             curLeft = parseFloat( curCSSLeft ) || 0;
10460         }
10461
10462         if ( jQuery.isFunction( options ) ) {
10463
10464             // Use jQuery.extend here to allow modification of coordinates argument (gh-1848)
10465             options = options.call( elem, i, jQuery.extend( {}, curOffset ) );
10466         }
10467
10468         if ( options.top != null ) {
10469             props.top = ( options.top - curOffset.top ) + curTop;
10470         }
10471         if ( options.left != null ) {
10472             props.left = ( options.left - curOffset.left ) + curLeft;
10473         }
10474
10475         if ( "using" in options ) {
10476             options.using.call( elem, props );
10477         }
10478         else {
10479             curElem.css( props );
10480         }
10481     }
10482 };
10483
10484 jQuery.fn.extend( {
10485
10486     // offset() relates an element's border box to the document origin
10487     offset: function( options ) {
10488
10489         // Preserve chaining for setter
10490         if ( arguments.length ) {
10491             return options === undefined ?
10492                 this :
10493                 this.each( function( i ) {
10494                     jQuery.offset.setOffset( this, options, i );
10495                 } );
10496         }

```

```

10497
10498 var rect, win;
10499 elem = this[ 0 ];
10500
10501 if ( !elem ) {
10502     return;
10503 }
10504
10505 // Return zeros for disconnected and hidden (display: none) elements (gh-2310)
10506 // Support: IE <=11 only
10507 // Running getBoundingClientRect on a
10508 // disconnected node in IE throws an error
10509 if ( !elem.getBoundingClientRect().length ) {
10510     return { top: 0, left: 0 };
10511 }
10512
10513 // Get document-relative position by adding viewport scroll to viewport-relative gBCR
10514 rect = elem.getBoundingClientRect();
10515 win = elem.ownerDocument.defaultView;
10516 return {
10517     top: rect.top + win.pageYOffset,
10518     left: rect.left + win.pageXOffset
10519 };
10520 },
10521
10522 // position() relates an element's margin box to its offset parent's padding box
10523 // This corresponds to the behavior of CSS absolute positioning
10524 position: function() {
10525     if ( !this[ 0 ] ) {
10526         return;
10527     }
10528
10529     var offsetParent, offset, doc;
10530     elem = this[ 0 ];
10531     parentOffset = { top: 0, left: 0 };
10532
10533     // position:fixed elements are offset from the viewport, which itself always has zero offset
10534     if ( jQuery.css( elem, "position" ) === "fixed" ) {
10535
10536         // Assume position:fixed implies availability of getBoundingClientRect
10537         offset = elem.getBoundingClientRect();
10538
10539     } else {
10540         offset = this.offset();
10541
10542         // Account for the "real" offset parent, which can be the document or its root element
10543         // when a statically positioned element is identified
10544         doc = elem.ownerDocument;
10545         offsetParent = elem.offsetParent || doc.documentElement;
10546         while ( ( offsetParent && offsetParent !== doc.body && offsetParent !== doc.documentElement ) &&
10547             jQuery.css( offsetParent, "position" ) === "static" ) {
10548             offsetParent = offsetParent.parentNode;
10549         }
10550
10551     }
10552     if ( offsetParent && offsetParent !== elem && offsetParent.nodeType === 1 ) {
10553
10554         // Incorporate borders into its offset, since they are outside its content origin
10555         parentOffset = jQuery( offsetParent ).offset();
10556         parentOffset.top += jQuery.css( offsetParent, "borderTopWidth", true );
10557         parentOffset.left += jQuery.css( offsetParent, "borderLeftWidth", true );
10558     }
10559 }
10560
10561 // Subtract parent offsets and element margins
10562 return {
10563     top: offset.top - parentOffset.top - jQuery.css( elem, "marginTop", true ),
10564     left: offset.left - parentOffset.left - jQuery.css( elem, "marginLeft", true )
10565 };
10566 },
10567
10568 // This method will return documentElement in the following cases:
10569 // 1) For the element inside the iframe without offsetParent, this method will return
10570 //    documentElement of the parent window
10571 // 2) For the hidden or detached element
10572 // 3) For body or html element, i.e. in case of the html node - it will return itself
10573 //
10574 // but those exceptions were never presented as a real life use-cases
10575 // and might be considered as more preferable results.
10576 //
10577 // This logic, however, is not guaranteed and can change at any point in the future
10578 offsetParent: function() {
10579     return this.map( function() {
10580         var offsetParent = this.offsetParent;
10581
10582         while ( offsetParent && jQuery.css( offsetParent, "position" ) === "static" ) {
10583             offsetParent = offsetParent.offsetParent;
10584         }
10585
10586         return offsetParent || documentElement;
10587     } );
10588 },
10589 );
10590
10591 // Create scrollLeft and scrollTop methods
10592 jQuery.each( { scrollLeft: "pageXOffset", scrollTop: "pageYOffset" }, function( method, prop ) {
10593     var top = "pageYOffset" === prop;
10594
10595     jQuery.fn[ method ] = function( val ) {
10596         return access( this, function( elem, method, val ) {
10597
10598             // Coalesce documents and windows
10599             var win;
10600             if ( !isWindow( elem ) ) {
10601                 win = elem;
10602             } else if ( !elem.nodeType === 9 ) {
10603                 win = elem.defaultView;
10604             }
10605
10606             if ( val === undefined ) {
10607                 return win ? win[ prop ] : elem[ method ];
10608             }
10609
10610             if ( win ) {
10611                 win.scrollTo(

```

```

10612     top ? val : win.pageXOffset,
10613     top ? val : win.pageYOffset
10614 );
10615
10616 } else {
10617     elem[ method ] = val;
10618 }
10619 }, method, val, arguments.length );
10620 };
10621 };
10622
10623 // Support: Safari <=7 - 9.1, Chrome <=37 - 49
10624 // Add the top/left cssHooks using jQuery.fn.position
10625 // Webkit bug: https://bugs.webkit.org/show_bug.cgi?id=29084
10626 // Blink bug: https://bugs.chromium.org/p/chromium/issues/detail?id=589347
10627 // getComputedStyle returns percent when specified for top/left/bottom/right;
10628 // rather than make the css module depend on the offset module, just check for it here
10629 jQuery.each( [ "top", "left" ], function( _i, prop ) {
10630     jQuery.cssHooks[ prop ] = addGetHookIf( support.pixelPosition,
10631         function( elem, computed ) {
10632             if ( computed ) {
10633                 computed = curCSS( elem, prop );
10634             }
10635             // If curCSS returns percentage, fallback to offset
10636             return rnumnonpx.test( computed ) ?
10637                 jQuery( elem ).position()[ prop ] + "px" :
10638                 computed;
10639         }
10640     );
10641 });
10642 };
10643
10644
10645 // Create innerHeight, innerWidth, height, width, outerHeight and outerWidth methods
10646 jQuery.each( { Height: "height", Width: "width" }, function( name, type ) {
10647     jQuery.each( [
10648         "padding" + name,
10649         "content" + name,
10650         "", "outer" + name
10651     ], function( defaultExtra, funcName ) {
10652         // Margin is only for outerHeight, outerWidth
10653         jQuery.fn[ funcName ] = function( margin, value ) {
10654             var chainable = arguments.length && { defaultExtra: true } || typeOf margin !== "boolean",
10655                 extra = defaultExtra || ! margin === true || value === true ? "margin" : "border";
10656
10657             return access( this, function( elem, type, value ) {
10658                 var doc;
10659
10660                 if ( isWindow( elem ) ) {
10661                     // $( window ).outerWidth/Height return w/h including scrollbars (gh-1729)
10662                     return funcName.indexOf( "outer" ) === 0 ?
10663                         elem[ "inner" + name ] :
10664                         elem.document.documentElement[ "client" + name ];
10665                 }
10666
10667                 // Get document width or height
10668                 if ( ! elem.nodeType ) {
10669                     doc = elem.document;
10670                 }
10671
10672                 // Either scroll[Width/Height] or offset[Width/Height] or client[Width/Height],
10673                 // whichever is greatest
10674                 return Math.max(
10675                     elem.body[ "scroll" + name ], doc[ "scroll" + name ],
10676                     elem.body[ "offset" + name ], doc[ "offset" + name ],
10677                     doc[ "client" + name ]
10678                 );
10679             }, type, chainable ? margin : undefined, chainable );
10680         };
10681     });
10682     return value === undefined ?
10683         // Get width or height on the element, requesting but not forcing parseFloat
10684         jQuery.css( elem, type, extra ) :
10685         jQuery( elem ).type( chainable ? margin : undefined, chainable );
10686 };
10687
10688 // Set width or height on the element
10689 jQuery.style( elem, type, value, extra );
10690
10691 // Set width or height on the element
10692 jQuery( elem ).type( chainable ? margin : undefined, chainable );
10693
10694
10695 jQuery.each( [
10696     "ajaxStart",
10697     "ajaxStop",
10698     "ajaxComplete",
10699     "ajaxError",
10700     "ajaxSuccess",
10701     "ajaxSend"
10702 ], function( _i, type ) {
10703     jQuery.fn[ type ] = function( fn ) {
10704         return this.on( type, fn );
10705     };
10706 });
10707
10708
10709
10710
10711 jQuery.fn.extend( {
10712     bind: function( types, data, fn ) {
10713         return this.on( types, null, data, fn );
10714     },
10715     unbind: function( types, fn ) {
10716         return this.off( types, null, fn );
10717     },
10718     delegate: function( selector, types, data, fn ) {
10719         return this.on( types, selector, data, fn );
10720     },
10721     undelegate: function( selector, types, fn ) {
10722         // ( namespace ) or ( selector, types [, fn] )
10723         return arguments.length === 1 ?

```

```

10727 this.off( selector, ".*" );
10728 this.off( types.selector || ".*", fn );
10729 },
10730
10731 hover: function( fnOver, fnOut ) {
10732     return this.mouseenter( fnOver ).mouseleave( fnOut || fnOver );
10733 }
10734 };
10735
10736 jQuery.each(
10737     [ "blur", "focus", "focusin", "focusout", "resize", "scroll", "click", "dblclick", " +
10738     "mousedown", "mouseup", "mousemove", "mouseover", "mouseout", "mouseenter", "mouseleave", " +
10739     "change", "select", "submit", "keydown", "keypress", "keyup", "contextmenu" ].split( " " ),
10740     function( _i, name ) {
10741
10742         // Handle event binding
10743         jQuery.fn[ name ] = function( data, fn ) {
10744             return arguments.length > 0 ?
10745                 this.on( name, null, data, fn ) :
10746                 this.trigger( name );
10747         };
10748     }
10749 );
10750
10751
10752
10753
10754 // Support: Android <=4.0 only
10755 // Make sure we trim BOM and NBSP
10756 var rtrim = /^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g;
10757
10758 // Bind a function to a context, optionally partially applying any
10759 // arguments.
10760 // jQuery.proxy is deprecated to promote standards (specifically Function#bind)
10761 // However, it is not slated for removal any time soon
10762 jQuery.proxy = function( fn, context ) {
10763     var tmp, args, proxy;
10764
10765     if ( !typeof context === "string" ) {
10766         tmp = fn; context = null;
10767     }
10768     fn = tmp;
10769
10770
10771     // Quick check to determine if target is callable, in the spec
10772     // this throws a TypeError, but we will just return undefined.
10773     if ( !isFunction( fn ) ) {
10774         return undefined;
10775     }
10776
10777     // Simulated bind
10778     args = slice.call( arguments, 2 );
10779     proxy = function() {
10780         return fn.apply( context || this, args.concat( slice.call( arguments ) ) );
10781     };
10782
10783     // Set the guid of unique handler to the same of original handler, so it can be removed
10784     proxy.guid = fn.guid = fn.guid || jQuery.guid++;
10785
10786     return proxy;
10787 };
10788
10789 jQuery.holdReady = function( hold ) {
10790     if ( hold ) {
10791         jQuery.readyWait++;
10792     } else {
10793         jQuery.ready( true );
10794     }
10795 };
10796
10797 jQuery.isArray = Array.isArray;
10798 jQuery.parseJSON = JSON.parse;
10799 jQuery.nodeName = nodeName;
10800 jQuery.isFunction = isFunction;
10801 jQuery.isWindow = isWindow;
10802 jQuery.camelCase = camelCase;
10803
10804 jQuery.type = toType;
10805
10806 jQuery.now = Date.now;
10807
10808 jQuery.isNumeric = function( obj ) {
10809     // As of jQuery 3.0, isNumeric is limited to
10810     // strings and numbers (primitives or objects)
10811     // that can be coerced to finite numbers (gh-2662)
10812     var type = jQuery.type( obj );
10813     return ( type === "number" || type === "string" ) &&
10814         // parseFloat NaNs numeric-cast false positives (""
10815         // ...but misinterprets leading-number strings, particularly hex literals ("0x...")
10816         // subtraction forces infinities to NaN
10817         !isNaN( obj - parseFloat( obj ) );
10818 };
10819
10820 jQuery.trim = function( text ) {
10821     return text == null ?
10822         "" :
10823         ( text + "" ).replace( rtrim, "" );
10824 };
10825
10826
10827
10828 // Register as a named AMD module, since jQuery can be concatenated with other
10829 // files that may use define, but not via a proper concatenation script that
10830 // understands anonymous AMD modules. A named AMD is safest and most robust
10831 // way to register. Lowercase jquery is used because AMD module names are
10832 // derived from file names, and jQuery is normally delivered in a lowercase
10833 // file name. Do this after creating the global so that if an AMD module wants
10834 // to call noConflict to hide this version of jQuery, it will work.
10835
10836 // Note that for maximum portability, libraries that are not jQuery should
10837 // declare themselves as anonymous modules, and avoid setting a global if an
10838 // AMD loader is present. jQuery is a special case. For more information, see
10839 // https://github.com/jrburke/requirejs/wiki/Updating-existing-libraries#wiki-anon
10840
10841 if ( typeof define === "function" && define.amd ) {

```

```
10842 define( "jquery", [], function() {
10843     return jQuery;
10844 } );
10845
10846
10847
10848
10849
10850 var
10851
10852 // Map over jQuery in case of overwrite
10853 _jQuery = window jQuery;
10854
10855 // Map over the $ in case of overwrite
10856 _$ = window $;
10857
10858 jQuery.noConflict = function( deep ) {
10859     if ( window $ === jQuery ) {
10860         window $ = _$;
10861     }
10862
10863     if ( deep && window jQuery === jQuery ) {
10864         window jQuery = _jQuery;
10865     }
10866
10867     return jQuery;
10868 };
10869
10870 // Expose jQuery and $ identifiers, even in AMD
10871 // (#7102#comment:10, https://github.com/jquery/jquery/pull/557)
10872 // and CommonJS for browser emulators (#13566)
10873 if ( typeof noGlobal === "undefined" ) {
10874     window jQuery = window $ = jQuery;
10875 }
10876
10877
10878
10879
10880 return jQuery;
10881 } );
```

**File: ./static/admin/js/vendor/jquery/jquery.min.js**

```
1 /*! jQuery v3.6.0 | (c) OpenJS Foundation and other contributors | jquery.org/license */
2 function(e,t){("use strict";"object"!==typeof module&&"object"!==typeof module.exports?module.exports=e:document?t(e,!0):function(e){if(!e)document.throw new Error("jQuery requires a window")}}
```

**File: ./static/admin/js/vendor/jquery/LICENSE.txt**

Copyright OpenJS Foundation and other contributors, <https://openjsf.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**File: ./static/admin/js/vendor/xregexp/xregexp.min.js**

```
1 /*
2  XRegExp.build 3.2.0
3  <xregexp.com>
4  Steven Levithan (c) 2012-2017 MIT License
5  Inspired by Lea Verou's RegExp.create <lea.verou.me>
6  XRegExp.matchRecursive 3.2.0
7  <xregexp.com>
8  Steven Levithan (c) 2009-2017 MIT License
9  XRegExp Unicode Base 3.2.0
10 <xregexp.com>
11 Steven Levithan (c) 2008-2017 MIT License
12 XRegExp Unicode Blocks 3.2.0
13 <xregexp.com>
14 Steven Levithan (c) 2010-2017 MIT License
15 Unicode data by Mathias Bynens <mathiasbynens.be>
16 XRegExp Unicode Categories 3.2.0
17 <xregexp.com>
18 Steven Levithan (c) 2010-2017 MIT License
19 Unicode data by Mathias Bynens <mathiasbynens.be>
20 XRegExp Unicode Properties 3.2.0
21 <xregexp.com>
22 Steven Levithan (c) 2012-2017 MIT License
23 Unicode data by Mathias Bynens <mathiasbynens.be>
```

[illegible]



[illegible]

```

88 * XRegExp.exec('10:59', time).minutes; // -> '59'
89 */
90 XRegExp.build = function(pattern, subs, flags) {
91   flags = flags || '';
92   // Used with `asXRegExp` calls for `pattern` and subpatterns in `subs`, to work around how
93   // some browsers convert `RegExp("\n")` to a regex that contains the literal characters `\`
94   // and `n`. See more details at <https://github.com/sievhithan/xregexp/pull/163>.
95   var addFlagX = flags.indexOf('x') > -1;
96   var inlineFlags = /^[^\\(\\?([\\w$]+)]/ exec(pattern);
97   // Add flags within a leading mode modifier to the overall pattern's flags
98   if (inlineFlags) {
99     flags = XRegExp._clipDuplicates(flags + inlineFlags[1]);
100   }
101
102   var data = {};
103   for (var p in subs) {
104     if (subs.hasOwnProperty(p)) {
105       // Passing to XRegExp enables extended syntax and ensures independent validity,
106       // lest an unescaped `(`, `)`, `[`, or trailing `\\` breaks the `(?:)` wrapper. For
107       // subpatterns provided as native regexes, it dies on octals and adds the property
108       // used to hold extended regex instance data, for simplicity.
109       var sub = asXRegExp(subs[p], addFlagX);
110       data[p] = {
111         // Deanchoring allows embedding independently useful anchored regexes. If you
112         // really need to keep your anchors, double them (i.e., `^...$`).
113         pattern: deanchor(sub.source),
114         names: sub.REGEX_DATA ? captureNames || []
115       };
116     }
117   }
118
119   // Passing to XRegExp dies on octals and ensures the outer pattern is independently valid;
120   // helps keep this simple. Named captures will be put back.
121   var patternAsRegex = asXRegExp(pattern, addFlagX);
122
123   // 'Caps' is short for 'captures'
124   var numCaps = 0;
125   var numPriorCaps;
126   var numOuterCaps = 0;
127   var outerCapsMap = [0];
128   var outerCapNames = patternAsRegex.REGEX_DATA ? captureNames || [];
129   var output = patternAsRegex.source.replace(patterns, function($0, $1, $2, $3, $4) {
130     var subName = $1 || $2;
131     var capName;
132     var intro;
133     var localCapIndex;
134     // Named subpattern
135     if (subName) {
136       if (!data.hasOwnProperty(subName)) {
137         throw new ReferenceError('Undefined property ' + $0);
138       }
139       // Named subpattern was wrapped in a capturing group
140       if ($1) {
141         capName = outerCapNames[numOuterCaps];
142         outerCapsMap[numOuterCaps] = ++numCaps;
143         // If it's a named group, preserve the name. Otherwise, use the subpattern name
144         // as the capture name
145         intro = '(?<' + (capName || subName) + '>';
146       } else {
147         intro = '(?:';
148       }
149       numPriorCaps = numCaps;
150       return intro + data[subName].pattern.replace(subParts, function(match, paren, backref) {
151         // Capturing group
152         if (paren) {
153           capName = data[subName].names[numCaps - numPriorCaps];
154           ++numCaps;
155           // If the current capture has a name, preserve the name
156           if (capName) {
157             return '(?<' + capName + '>';
158           }
159           // Backreference
160         } else if (backref) {
161           localCapIndex = +backref - 1;
162           // Rewrite the backreference
163           return data[subName].names[localCapIndex] ?
164             // Need to preserve the backreference name in case using flag `n`
165             '\\k<' + data[subName].names[localCapIndex] + '>' :
166             '\\1' + (+backref + numPriorCaps);
167         }
168         return match;
169       }) + ')';
170     }
171     // Capturing group
172     if ($3) {
173       capName = outerCapNames[numOuterCaps];
174       outerCapsMap[numOuterCaps] = ++numCaps;
175       // If the current capture has a name, preserve the name
176       if (capName) {
177         return '(?<' + capName + '>';
178       }
179       // Backreference
180     } else if ($4) {
181       localCapIndex = +$4 - 1;
182       // Rewrite the backreference
183       return outerCapNames[localCapIndex] ?
184         // Need to preserve the backreference name in case using flag `n`
185         '\\k<' + outerCapNames[localCapIndex] + '>' :
186         '\\1' + outerCapsMap[numOuterCaps];
187     }
188     return $0;
189   });
190   return XRegExp(output, flags);
191 };
192
193
194 };
195
196 }, {}), 2: (function(require, module, exports) {
197   /*
198   * XRegExp.matchRecursive 3.2.0
199   * <xregexp.com>
200   * Steven Levithan (c) 2009-2017 MIT License
201   */
202

```

```

203 module exports = function XRegExp (
204   'use strict';
205
206   /**
207    * Returns a match detail object composed of the provided values.
208    *
209    * @private
210    */
211   function row(name, value, start, end) {
212     return {
213       name: name,
214       value: value,
215       start: start,
216       end: end
217     };
218   }
219
220   /**
221    * Returns an array of match strings between outermost left and right delimiters, or an array of
222    * objects with detailed match parts and position data. An error is thrown if delimiters are
223    * unbalanced within the data.
224    *
225    * @memberOf XRegExp
226    * @param {String} str String to search.
227    * @param {String} left Left delimiter as an XRegExp pattern.
228    * @param {String} right Right delimiter as an XRegExp pattern.
229    * @param {String} [flags] Any native or XRegExp flags, used for the left and right delimiters.
230    * @param {Object} [options] Lets you specify `valueNames` and `escapeChar` options.
231    * @returns {Array} Array of matches, or an empty array.
232    * @example
233    *
234    * // Basic usage
235    * var str = 't((e))s)t((ing)';
236    * XRegExp.matchRecursive(str, '\\(', '\\)', 'g');
237    * // -> ['t((e))s', '', 'ing']
238    *
239    * // Extended information mode with valueNames
240    * str = 'Here is <div> <div>an</div></div> example';
241    * XRegExp.matchRecursive(str, '<div\\s*>', '</div>', 'gi', {
242    *   valueNames: ['between', 'left', 'match', 'right']
243    * });
244    * // -> [
245    * // {name: 'between', value: 'Here is ', start: 0, end: 8},
246    * // {name: 'left', value: '<div>', start: 8, end: 13},
247    * // {name: 'match', value: ' <div>an</div>', start: 13, end: 27},
248    * // {name: 'right', value: '</div>', start: 27, end: 33},
249    * // {name: 'between', value: ' example', start: 33, end: 41}
250    * // ]
251    *
252    * // Omitting unneeded parts with null valueNames, and using escapeChar
253    * str = '...{1}.\\{function(x,y){return {y:x}}}'
254    * XRegExp.matchRecursive(str, '{', '}', 'g', {
255    *   valueNames: ['literal', null, 'value', null],
256    *   escapeChar: '\\'
257    * });
258    * // -> [
259    * // {name: 'literal', value: '...', start: 0, end: 3},
260    * // {name: 'value', value: '1', start: 4, end: 5},
261    * // {name: 'literal', value: '\\{', start: 6, end: 9},
262    * // {name: 'value', value: 'function(x,y){return {y:x}}', start: 10, end: 37}
263    * // ]
264    *
265    * // Sticky mode via flag y
266    * str = '<1><<2>>><3>4<5>';
267    * XRegExp.matchRecursive(str, '<', '>', 'gy');
268    * // -> ['1', '<<2>>', '3']
269    */
270   XRegExp.matchRecursive = function(str, left, right, flags, options) {
271     flags = flags || '';
272     options = options || {};
273     var global = flags.indexOf('g') > -1;
274     var sticky = flags.indexOf('y') > -1;
275     // Flag `y` is controlled internally
276     var basicFlags = flags.replace(/y/g, '');
277     var escapeChar = options.escapeChar;
278     var vN = options.valueNames;
279     var output = [];
280     var openTokens = 0;
281     var delimStart = 0;
282     var delimEnd = 0;
283     var lastOuterEnd = 0;
284     var outerStart;
285     var innerStart;
286     var leftMatch;
287     var rightMatch;
288     var esc;
289     left = XRegExp(left, basicFlags);
290     right = XRegExp(right, basicFlags);
291
292     if (escapeChar) {
293       if (escapeChar.length > 1) {
294         throw new Error('Cannot use more than one escape character');
295       }
296       escapeChar = XRegExp.escape(escapeChar);
297       // Example of concatenated `esc` regex:
298       // `escapeChar`: '%'
299       // `left`: '<'
300       // `right`: '>'
301       // Regex is: /(?:%[\Ss])[(?:?!<|>)[^%])+)/
302       esc = new RegExp(
303         '(?:' + escapeChar + '[\\S\\s])[(?:?!' +
304         // Using `XRegExp.union` safely rewrites backreferences in `left` and `right`.
305         // Intentionally not passing `basicFlags` to `XRegExp.union` since any syntax
306         // transformation resulting from those flags was already applied to `left` and
307         // `right` when they were passed through the XRegExp constructor above.
308         XRegExp.union([left, right], {conjunction: 'or'}) source +
309         ')(^|' + escapeChar + '))+)',
310         // Flags `gy` not needed here
311         flags.replace(/([imuyg])/g, '')
312       );
313     }
314
315     while (true) {
316       // If using an escape character, advance to the delimiter's next starting position,
317       // skipping any escaped characters in between

```

```

318     if (escapeChar) {
319         delimEnd += XRegExp.exec(str, esc, delimEnd, 'sticky') || [''])[0].length;
320     }
321     leftMatch = XRegExp.exec(str, left, delimEnd);
322     rightMatch = XRegExp.exec(str, right, delimEnd);
323     // Keep the leftmost match only
324     if (leftMatch && rightMatch) {
325         if (leftMatch.index <= rightMatch.index) {
326             rightMatch = null;
327         } else {
328             leftMatch = null;
329         }
330     }
331     // Paths (LM: leftMatch, RM: rightMatch, OT: openTokens):
332     // LM | RM | OT | Result
333     // 1 | 0 | 1 | loop
334     // 1 | 0 | 0 | loop
335     // 0 | 1 | 1 | loop
336     // 0 | 1 | 0 | throw
337     // 0 | 0 | 1 | throw
338     // 0 | 0 | 0 | break
339     // The paths above don't include the sticky mode special case. The loop ends after the
340     // first completed match if not 'global'.
341     if (leftMatch || rightMatch) {
342         delimStart = (leftMatch || rightMatch).index;
343         delimEnd = delimStart + (leftMatch || rightMatch)[0].length;
344     } else if (!openTokens) {
345         break;
346     }
347     if (sticky && openTokens && delimStart > lastOuterEnd) {
348         break;
349     }
350     if (leftMatch) {
351         if (!openTokens) {
352             outerStart = delimStart;
353             innerStart = delimEnd;
354         }
355         ++openTokens;
356     } else if (rightMatch && openTokens) {
357         if (!--openTokens) {
358             if (vN) {
359                 if (vN[0] && outerStart > lastOuterEnd) {
360                     output.push(row.vN[0], str.slice(lastOuterEnd, outerStart), lastOuterEnd, outerStart);
361                 }
362                 if (vN[1]) {
363                     output.push(row.vN[1], str.slice(outerStart, innerStart), outerStart, innerStart);
364                 }
365                 if (vN[2]) {
366                     output.push(row.vN[2], str.slice(innerStart, delimStart), innerStart, delimStart);
367                 }
368                 if (vN[3]) {
369                     output.push(row.vN[3], str.slice(delimStart, delimEnd), delimStart, delimEnd);
370                 }
371             } else {
372                 output.push(str.slice(innerStart, delimStart));
373             }
374             lastOuterEnd = delimEnd;
375             if (!global) {
376                 break;
377             }
378         }
379     } else {
380         throw new Error("Unbalanced delimiter found in string");
381     }
382     // If the delimiter matched an empty string, avoid an infinite loop
383     if (delimStart === delimEnd) {
384         ++delimEnd;
385     }
386 }
387
388 if (global && !sticky && vN && vN[0] && str.length > lastOuterEnd) {
389     output.push(row.vN[0], str.slice(lastOuterEnd, lastOuterEnd, str.length));
390 }
391
392 return output;
393 };
394
395 };
396
397 },{}],3:(function(require,module,exports){
398     /*
399     * XRegExp Unicode Base 3.2.0
400     * <xregexp.com>
401     * Steven Levithan (c) 2008-2017 MIT License
402     */
403
404     module.exports = function(XRegExp) {
405         'use strict';
406
407         /**
408         * Adds base support for Unicode matching:
409         * - Adds syntax `p{...}` for matching Unicode tokens. Tokens can be inverted using `P{...}` or
410         *   `p{^...}`. Token names ignore case, spaces, hyphens, and underscores. You can omit the
411         *   braces for token names that are a single letter (e.g. `pL` or `PL`).
412         * - Adds flag A (astral), which enables 21-bit Unicode support.
413         * - Adds the `XRegExp.addUnicodeData` method used by other addons to provide character data.
414         *
415         * Unicode Base relies on externally provided Unicode character data. Official addons are
416         * available to provide data for Unicode categories, scripts, blocks, and properties.
417         *
418         * @requires XRegExp
419         */
420
421         // ==-----
422         // Private stuff
423         // ==-----
424
425         // Storage for Unicode data
426         var unicode = {},
427
428         // Reuse utils
429         var dec = XRegExp._dec,
430             var hex = XRegExp._hex,
431             var pad4 = XRegExp._pad4;
432

```

```

431 // Generates a token lookup name: lowercase, with hyphens, spaces, and underscores removed
432 function normalize(name) {
433     return name.replace(/[-_+ /g, " ") toLowerCase;
434 }
435
436 // Gets the decimal code of a literal code unit, \xHH, \uHHHH, or a backslash-escaped literal
437 function charCode(chr) {
438     var esc = /^\\[xu](.+)/; exec: chr;
439     return esc ?
440         dec: esc[1] :
441         chr.charCodeAt(0) === '\0' ? 1 : 0;
442 }
443
444 // Inverts a list of ordered BMP characters and ranges
445 function invertBmp(range) {
446     var output = "";
447     var lastEnd = -1;
448
449     XRegExp.forEach(
450         range
451         /\x..\u....(?:[sS])(?:-[x...]\u....(?:[sS]))?/,
452         function(m) {
453             var start = charCode(m[1]);
454             if (start > (lastEnd + 1)) {
455                 output += '\u' + pad4(hex(lastEnd + 1));
456                 if (start > (lastEnd + 2)) {
457                     output += '\u' + pad4(hex(start - 1));
458                 }
459             }
460             lastEnd = charCode(m[2] || m[1]);
461         }
462     );
463
464     if (lastEnd < 0xFFFF) {
465         output += '\u' + pad4(hex(lastEnd + 1));
466         if (lastEnd < 0xFFFF) {
467             output += '\uFFFF';
468         }
469     }
470
471     return output;
472 }
473
474 // Generates an inverted BMP range on first use
475 function cacheInvertedBmp(slug) {
476     var prop = 'b!';
477     return (
478         unicode.slug || prop ||
479         (unicode.slug || prop) = invertBmp(unicode.slug || bmp)
480     );
481 }
482
483 // Combines and optionally negates BMP and astral data
484 function buildAstral(slug, isNegated) {
485     var item = unicode.slug;
486     var combined = "";
487
488     if (item.bmp && !item.isBmpLast) {
489         combined = '[' + item.bmp + ']' + (item.astral ? '[' : "");
490     }
491     if (item.astral) {
492         combined += item.astral;
493     }
494     if (item.isBmpLast && item.bmp) {
495         combined += (item.astral ? '[' : "") + '[' + item.bmp + ']'
496     }
497
498     // Astral Unicode tokens always match a code point, never a code unit
499     return isNegated ?
500         '^(?:' + combined + ')(?:[\uDB80-\uDBFF][\uDC00-\uDFFF][\u0000-\uFFFF])' :
501         '(?:' + combined + ')';
502 }
503
504 // Builds a complete astral pattern on first use
505 function cacheAstral(slug, isNegated) {
506     var prop = isNegated ? 'a!' : 'a=';
507     return (
508         unicode.slug || prop ||
509         (unicode.slug || prop) = buildAstral(slug, isNegated)
510     );
511 }
512
513 // =====
514 // Core functionality
515 // =====
516
517 /*
518 * Add astral mode (flag A) and Unicode token syntax: '\p{...}', '\P{...}', '\p{^...}', '\pC'.
519 */
520
521 XRegExp.addToken(
522     // Use '*' instead of '^' to avoid capturing '^' as the token name in '\p{^}'
523     /\([pP](?:\{(?:\^?)\{(?:\^?)\}|\{[A-Za-z]\})\})/,
524     function(match, scope, flags) {
525         var ERR_DOUBLE_NEG = 'Invalid double negation';
526         var ERR_UNKNOWN_NAME = 'Unknown Unicode token';
527         var ERR_UNKNOWN_REF = 'Unicode token missing data';
528         var ERR_ASTRAL_ONLY = 'Astral mode required for Unicode token';
529         var ERR_ASTRAL_IN_CLASS = 'Astral mode does not support Unicode tokens within character classes';
530         // Negated via '\P{...}' or '\p{^...}'
531         var isNegated = match[1] === 'P' || !match[2];
532         // Switch from BMP (0-FFFF) to astral (0-10FFFF) mode via flag A
533         var isAstralMode = flags.indexOf('A') > -1;
534         // Token lookup name. Check '[4]' first to avoid passing 'undefined' via '\p{'
535         var slug = normalize(match[4] || match[3]);
536         // Token data object
537         var item = unicode.slug;
538
539         if (match[1] === 'P' && match[2]) {
540             throw new SyntaxError(ERR_DOUBLE_NEG + match[0]);
541         }
542         if (!unicode.hasOwnProperty(slug)) {
543             throw new SyntaxError(ERR_UNKNOWN_NAME + match[0]);
544         }
545
546         // Switch to the negated form of the referenced Unicode token
547     }
548 );

```

```

548     if (item.inverseOf) {
549         slug = normalize(item.inverseOf);
550         if (!unicode.hasOwnProperty(slug)) {
551             throw new ReferenceError(ERR_UNKNOWN_REF + match[0] + ' -> ' + item.inverseOf);
552         }
553         item = unicode[slug];
554         isNegated = !isNegated;
555     }
556
557     if (!(item.bmp || isAstralMode)) {
558         throw new SyntaxError(ERR_ASTRAL_ONLY + match[0]);
559     }
560     if (isAstralMode) {
561         if (scope === 'class') {
562             throw new SyntaxError(ERR_ASTRAL_IN_CLASS);
563         }
564
565         return cacheAstral(slug, isNegated);
566     }
567
568     return scope === 'class' ?
569         (isNegated ? cacheInvertedBmp(slug) : item.bmp) :
570         (isNegated ? '[' + item.bmp + ']' :
571     );
572     {
573         scope: 'all',
574         optionalFlags: 'A',
575         leadChar: '\\
576     }
577 );
578
579 /**
580  * Adds to the list of Unicode tokens that XRegExp regexes can match via `p` or `P`.
581  *
582  * @memberOf XRegExp
583  * @param {Array} data Objects with named character ranges. Each object may have properties
584  *   `name`, `alias`, `isBmpLast`, `inverseOf`, `bmp`, and `astral`. All but `name` are
585  *   optional, although one of `bmp` or `astral` is required (unless `inverseOf` is set). If
586  *   `astral` is absent, the `bmp` data is used for BMP and astral modes. If `bmp` is absent,
587  *   the name errors in BMP mode but works in astral mode. If both `bmp` and `astral` are
588  *   provided, the `bmp` data only is used in BMP mode, and the combination of `bmp` and
589  *   `astral` data is used in astral mode. `isBmpLast` is needed when a token matches orphan
590  *   high surrogates *and* uses surrogate pairs to match astral code points. The `bmp` and
591  *   `astral` data should be a combination of literal characters and `\\xHH` or `\\uHHHH` escape
592  *   sequences, with hyphens to create ranges. Any regex metacharacters in the data should be
593  *   escaped, apart from range-creating hyphens. The `astral` data can additionally use
594  *   character classes and alternation, and should use surrogate pairs to represent astral code
595  *   points. `inverseOf` can be used to avoid duplicating character data if a Unicode token is
596  *   defined as the exact inverse of another token.
597  * @example
598  *
599  * // Basic use
600  * XRegExp.addUnicodeData([
601  *   { name: 'XDigit',
602  *     alias: 'Hexadecimal',
603  *     bmp: '0-9A-Fa-f'
604  * }]);
605  * XRegExp('\\p{XDigit}:\\p{Hexadecimal}+').test('0:3D'); // -> true
606  */
607 XRegExp.addUnicodeData = function(data) {
608     var ERR_NO_NAME = 'Unicode token requires name';
609     var ERR_NO_DATA = 'Unicode token has no character data';
610     var item;
611
612     for (var i = 0; i < data.length; ++i) {
613         item = data[i];
614         if (!item.name) {
615             throw new Error(ERR_NO_NAME);
616         }
617         if (!(item.inverseOf || item.bmp || item.astral)) {
618             throw new Error(ERR_NO_DATA + item.name);
619         }
620         unicode[normalize(item.name)] = item;
621         if (item.alias) {
622             unicode[normalize(item.alias)] = item;
623         }
624     }
625
626     // Reset the pattern cache used by the `XRegExp` constructor, since the same pattern and
627     // flags might now produce different results
628     XRegExp.cache.flush('patterns');
629 };
630
631 /**
632  * @ignore
633  *
634  * Return a reference to the internal Unicode definition structure for the given Unicode
635  * Property if the given name is a legal Unicode Property for use in XRegExp `p` or `P` regex
636  * constructs.
637  *
638  * @memberOf XRegExp
639  * @param {String} name Name by which the Unicode Property may be recognized (case-insensitive),
640  *   e.g. `\\N` or `\\Number`. The given name is matched against all registered Unicode
641  *   Properties and Property Aliases.
642  * @returns {Object} Reference to definition structure when the name matches a Unicode Property.
643  *
644  * @note
645  * For more info on Unicode Properties, see also http://unicode.org/reports/tr18/#Categories.
646  *
647  * @note
648  * This method is *not* part of the officially documented API and may change or be removed in
649  * the future. It is meant for userland code that wishes to reuse the (large) internal Unicode
650  * structures set up by XRegExp.
651  */
652 XRegExp._getUnicodeProperty = function(name) {
653     var slug = normalize(name);
654     return unicode[slug];
655 };
656
657 };
658
659 },{}],4:[function(require,module,exports){
660     /*!
661     * XRegExp Unicode Blocks 3.2.0
662     * <xregexp.com>

```

```

663 * Steven Levithan (c) 2010-2017 MIT License
664 * Unicode data by Mathias Bynens <mathiasbynens.be>
665 */
666
667 module exports = function(XRegExp) {
668   'use strict';
669
670   /**
671    * Adds support for all Unicode blocks. Block names use the prefix 'In'. E.g.,
672    * '\p{InBasicLatin}'. Token names are case insensitive, and any spaces, hyphens, and
673    * underscores are ignored.
674    *
675    * Uses Unicode 9.0.0.
676    *
677    * @requires XRegExp, Unicode Base
678    */
679
680   if (!XRegExp.addUnicodeData) {
681     throw new ReferenceError('Unicode Base must be loaded before Unicode Blocks');
682   }
683
684   XRegExp.addUnicodeData([
685     {
686       name: 'InAdlam',
687       astral: '\uD83A\uDD00-\uDD5F'
688     },
689     {
690       name: 'InAegean_Numbers',
691       astral: '\uD800\uDD00-\uDD3F'
692     },
693     {
694       name: 'InAhom',
695       astral: '\uD805\uDF00-\uDF3F'
696     },
697     {
698       name: 'InAlchemical_Symbols',
699       astral: '\uD83D\uDF00-\uDF7F'
700     },
701     {
702       name: 'InAlphabetic_Presentation_Forms',
703       bmp: '\uFB00-\uFB4F'
704     },
705     {
706       name: 'InAnatolian_Hieroglyphs',
707       astral: '\uD811\uDC00-\uDE7F'
708     },
709     {
710       name: 'InAncient_Greek_Musical_Notation',
711       astral: '\uD834\uDE00-\uDE4F'
712     },
713     {
714       name: 'InAncient_Greek_Numbers',
715       astral: '\uD800\uDD40-\uDD8F'
716     },
717     {
718       name: 'InAncient_Symbols',
719       astral: '\uD800\uDD90-\uDDCF'
720     },
721     {
722       name: 'InArabic',
723       bmp: '\u0600-\u06FF'
724     },
725     {
726       name: 'InArabic_Extended_A',
727       bmp: '\u08A0-\u08FF'
728     },
729     {
730       name: 'InArabic_Mathematical_Alphabetic_Symbols',
731       astral: '\uD83B\uDE00-\uDEFF'
732     },
733     {
734       name: 'InArabic_Presentation_Forms_A',
735       bmp: '\uFB50-\uFDF'
736     },
737     {
738       name: 'InArabic_Presentation_Forms_B',
739       bmp: '\uFE70-\uFEFF'
740     },
741     {
742       name: 'InArabic_Supplement',
743       bmp: '\u0750-\u077F'
744     },
745     {
746       name: 'InArmenian',
747       bmp: '\u0530-\u058F'
748     },
749     {
750       name: 'InArrows',
751       bmp: '\u2190-\u21FF'
752     },
753     {
754       name: 'InAvestan',
755       astral: '\uD802\uDF00-\uDF3F'
756     },
757     {
758       name: 'InBalinese',
759       bmp: '\u1B00-\u1B7F'
760     },
761     {
762       name: 'InBamum',
763       bmp: '\uA6A0-\uA6FF'
764     },
765     {
766       name: 'InBamum_Supplement',
767       astral: '\uD81A\uDC00-\uDE3F'
768     },
769     {
770       name: 'InBasic_Latin',
771       bmp: '\0-\x7F'
772     },
773     {
774       name: 'InBassa_Vah',
775       astral: '\uD81A\uDED0-\uDEFF'
776     },
777     {

```

```

778     name: 'InBatak',
779     bmp: '\u1BC0-\u1BFF'
780 },
781 {
782     name: 'InBengali',
783     bmp: '\u0980-\u09FF'
784 },
785 {
786     name: 'InBhaiksuki',
787     astral: '\uD807[\uDC00-\uDC6F]'
788 },
789 {
790     name: 'InBlock_Elements',
791     bmp: '\u2580-\u259F'
792 },
793 {
794     name: 'InBopomofo',
795     bmp: '\u3100-\u312F'
796 },
797 {
798     name: 'InBopomofo_Extended',
799     bmp: '\u31A0-\u31BF'
800 },
801 {
802     name: 'InBox_Drawing',
803     bmp: '\u2500-\u257F'
804 },
805 {
806     name: 'InBrahmi',
807     astral: '\uD804[\uDC00-\uDC7F]'
808 },
809 {
810     name: 'InBraille_Patterns',
811     bmp: '\u2800-\u28FF'
812 },
813 {
814     name: 'InBuginese',
815     bmp: '\u1A00-\u1A1F'
816 },
817 {
818     name: 'InBuhid',
819     bmp: '\u1740-\u175F'
820 },
821 {
822     name: 'InByzantine_Musical_Symbols',
823     astral: '\uD834[\uDC00-\uDCFF]'
824 },
825 {
826     name: 'InCJK_Compatibility',
827     bmp: '\u3300-\u33FF'
828 },
829 {
830     name: 'InCJK_Compatibility_Forms',
831     bmp: '\uFE30-\uFE4F'
832 },
833 {
834     name: 'InCJK_Compatibility_Ideographs',
835     bmp: '\uF900-\uFAFF'
836 },
837 {
838     name: 'InCJK_Compatibility_Ideographs_Supplement',
839     astral: '\uD87E[\uDC00-\uDE1F]'
840 },
841 {
842     name: 'InCJK_Radicals_Supplement',
843     bmp: '\u2E80-\u2EFF'
844 },
845 {
846     name: 'InCJK_Strokes',
847     bmp: '\u31C0-\u31EF'
848 },
849 {
850     name: 'InCJK_Symbols_and_Punctuation',
851     bmp: '\u3000-\u303F'
852 },
853 {
854     name: 'InCJK_Unified_Ideographs',
855     bmp: '\u4E00-\u9FFF'
856 },
857 {
858     name: 'InCJK_Unified_Ideographs_Extension_A',
859     bmp: '\u3400-\u4DBF'
860 },
861 {
862     name: 'InCJK_Unified_Ideographs_Extension_B',
863     astral: '\uD840-\uD868[\uDC00-\uDFFF]\uD869[\uDC00-\uDEDF]'
864 },
865 {
866     name: 'InCJK_Unified_Ideographs_Extension_C',
867     astral: '\uD869[\uDF00-\uDFFF][\uD86A-\uD86C][\uDC00-\uDFFF]\uD86D[\uDC00-\uDF3F]'
868 },
869 {
870     name: 'InCJK_Unified_Ideographs_Extension_D',
871     astral: '\uD86D[\uDF40-\uDFFF][\uD86E[\uDC00-\uDC1F]'
872 },
873 {
874     name: 'InCJK_Unified_Ideographs_Extension_E',
875     astral: '\uD86E[\uDC20-\uDFFF][\uD86F-\uD872][\uDC00-\uDFFF]\uD873[\uDC00-\uDEAF]'
876 },
877 {
878     name: 'InCarian',
879     astral: '\uD800[\uDEA0-\uDEDF]'
880 },
881 {
882     name: 'InCaucasian_Albanian',
883     astral: '\uD801[\uDD30-\uDD6F]'
884 },
885 {
886     name: 'InChakma',
887     astral: '\uD804[\uDD00-\uDD4F]'
888 },
889 {
890     name: 'InCham',
891     bmp: '\uAA00-\uAA5F'
892 },

```



```

893 {
894     name: 'InCherokee',
895     bmp: '\u13A0-\u13FF'
896 },
897 {
898     name: 'InCherokee_Supplement',
899     bmp: '\uAB70-\uABBF'
900 },
901 {
902     name: 'InCombining_Diacritical_Marks',
903     bmp: '\u0300-\u036F'
904 },
905 {
906     name: 'InCombining_Diacritical_Marks_Extended',
907     bmp: '\u1AB0-\u1AFF'
908 },
909 {
910     name: 'InCombining_Diacritical_Marks_Supplement',
911     bmp: '\u1DC0-\u1DFF'
912 },
913 {
914     name: 'InCombining_Diacritical_Marks_for_Symbols',
915     bmp: '\u20D0-\u20FF'
916 },
917 {
918     name: 'InCombining_Half_Marks',
919     bmp: '\uFE20-\uFE2F'
920 },
921 {
922     name: 'InCommon_Indic_Number_Forms',
923     bmp: '\uA830-\uA83F'
924 },
925 {
926     name: 'InControl_Pictures',
927     bmp: '\u2400-\u243F'
928 },
929 {
930     name: 'InCoptic',
931     bmp: '\u2C80-\u2CFF'
932 },
933 {
934     name: 'InCoptic_Epact_Numbers',
935     astral: '\uD800[\uDEE0-\uDEFF]'
936 },
937 {
938     name: 'InCounting_Rod_Numerals',
939     astral: '\uD834[\uDF60-\uDF7F]'
940 },
941 {
942     name: 'InCuneiform',
943     astral: '\uD808[\uDC00-\uDFFF]'
944 },
945 {
946     name: 'InCuneiform_Numbers_and_Punctuation',
947     astral: '\uD809[\uDC00-\uDC7F]'
948 },
949 {
950     name: 'InCurrency_Symbols',
951     bmp: '\u20A0-\u20CF'
952 },
953 {
954     name: 'InCypriot_Syllabary',
955     astral: '\uD802[\uDC00-\uDC3F]'
956 },
957 {
958     name: 'InCyrillic',
959     bmp: '\u0400-\u04FF'
960 },
961 {
962     name: 'InCyrillic_Extended_A',
963     bmp: '\u2DE0-\u2DFF'
964 },
965 {
966     name: 'InCyrillic_Extended_B',
967     bmp: '\uA640-\uA69F'
968 },
969 {
970     name: 'InCyrillic_Extended_C',
971     bmp: '\u1C80-\u1C8F'
972 },
973 {
974     name: 'InCyrillic_Supplement',
975     bmp: '\u0500-\u052F'
976 },
977 {
978     name: 'InDeseret',
979     astral: '\uD801[\uDC00-\uDC4F]'
980 },
981 {
982     name: 'InDevanagari',
983     bmp: '\u0900-\u097F'
984 },
985 {
986     name: 'InDevanagari_Extended',
987     bmp: '\uA8E0-\uA8FF'
988 },
989 {
990     name: 'InDingbats',
991     bmp: '\u2700-\u27BF'
992 },
993 {
994     name: 'InDomino_Tiles',
995     astral: '\uD83C[\uDC30-\uDC9F]'
996 },
997 {
998     name: 'InDuployan',
999     astral: '\uD82F[\uDC00-\uDC9F]'
1000 },
1001 {
1002     name: 'InEarly_Dynastic_Cuneiform',
1003     astral: '\uD809[\uDC80-\uDD4F]'
1004 },
1005 {
1006     name: 'InEgyptian_Hieroglyphs',
1007     astral: '\uD80C[\uDC00-\uDFFF][\uD80D[\uDC00-\uDC2F]'

```

```

1008 },
1009 {
1010     name: 'InElbasan',
1011     astral: '\uD801[\uDD00-\uDD2F]'
1012 },
1013 {
1014     name: 'InEmoticons',
1015     astral: '\uD83D[\uDE00-\uDE4F]'
1016 },
1017 {
1018     name: 'InEnclosed_Alphanumeric_Supplement',
1019     astral: '\uD83C[\uDD00-\uDFFF]'
1020 },
1021 {
1022     name: 'InEnclosed_Alphanumerics',
1023     bmp: '\u2460-\u24FF'
1024 },
1025 {
1026     name: 'InEnclosed_CJK_Letters_and_Months',
1027     bmp: '\u3200-\u32FF'
1028 },
1029 {
1030     name: 'InEnclosed_Ideographic_Supplement',
1031     astral: '\uD83C[\uDE00-\uDEFF]'
1032 },
1033 {
1034     name: 'InEthiopic',
1035     bmp: '\u1200-\u137F'
1036 },
1037 {
1038     name: 'InEthiopic_Extended',
1039     bmp: '\u2D80-\u2DDF'
1040 },
1041 {
1042     name: 'InEthiopic_Extended_A',
1043     bmp: '\uAB00-\uAB2F'
1044 },
1045 {
1046     name: 'InEthiopic_Supplement',
1047     bmp: '\u1380-\u139F'
1048 },
1049 {
1050     name: 'InGeneral_Punctuation',
1051     bmp: '\u2000-\u206F'
1052 },
1053 {
1054     name: 'InGeometric_Shapes',
1055     bmp: '\u25A0-\u25FF'
1056 },
1057 {
1058     name: 'InGeometric_Shapes_Extended',
1059     astral: '\uD83D[\uDF80-\uDFFF]'
1060 },
1061 {
1062     name: 'InGeorgian',
1063     bmp: '\u10A0-\u10FF'
1064 },
1065 {
1066     name: 'InGeorgian_Supplement',
1067     bmp: '\u2D00-\u2D2F'
1068 },
1069 {
1070     name: 'InGlagolitic',
1071     bmp: '\u2C00-\u2C5F'
1072 },
1073 {
1074     name: 'InGlagolitic_Supplement',
1075     astral: '\uD838[\uDC00-\uDC2F]'
1076 },
1077 {
1078     name: 'InGothic',
1079     astral: '\uD800[\uDF30-\uDF4F]'
1080 },
1081 {
1082     name: 'InGrantha',
1083     astral: '\uD804[\uDF00-\uDF7F]'
1084 },
1085 {
1086     name: 'InGreek_Extended',
1087     bmp: '\u1F00-\u1FFF'
1088 },
1089 {
1090     name: 'InGreek_and_Coptic',
1091     bmp: '\u0370-\u03FF'
1092 },
1093 {
1094     name: 'InGujarati',
1095     bmp: '\u0A80-\u0AFF'
1096 },
1097 {
1098     name: 'InGurmukhi',
1099     bmp: '\u0A00-\u0A7F'
1100 },
1101 {
1102     name: 'InHalfwidth_and_Fullwidth_Forms',
1103     bmp: '\uFF00-\uFFEF'
1104 },
1105 {
1106     name: 'InHangul_Compatibility_Jamo',
1107     bmp: '\u3130-\u318F'
1108 },
1109 {
1110     name: 'InHangul_Jamo',
1111     bmp: '\u1100-\u11FF'
1112 },
1113 {
1114     name: 'InHangul_Jamo_Extended_A',
1115     bmp: '\uA960-\uA97F'
1116 },
1117 {
1118     name: 'InHangul_Jamo_Extended_B',
1119     bmp: '\uD7B0-\uD7FF'
1120 },
1121 {
1122     name: 'InHangul_Syllables',

```

```

1123     bmp: '\uAC00-\uD7AF'
1124 },
1125 {
1126     name: 'InHanunoo',
1127     bmp: '\u1720-\u173F'
1128 },
1129 {
1130     name: 'InHatan',
1131     astral: '\uD802[\uDC00-\uDCFF]'
1132 },
1133 {
1134     name: 'InHebrew',
1135     bmp: '\u0590-\u05FF'
1136 },
1137 {
1138     name: 'InHigh_Private_Use_Surrogates',
1139     bmp: '\uDB80-\uDBFF'
1140 },
1141 {
1142     name: 'InHigh_Surrogates',
1143     bmp: '\uD800-\uDB7F'
1144 },
1145 {
1146     name: 'InHiragana',
1147     bmp: '\u3040-\u309F'
1148 },
1149 {
1150     name: 'InIPA_Extensions',
1151     bmp: '\u0250-\u02AF'
1152 },
1153 {
1154     name: 'InIdeographic_Description_Characters',
1155     bmp: '\u2FF0-\u2FFF'
1156 },
1157 {
1158     name: 'InIdeographic_Symbols_and_Punctuation',
1159     astral: '\uD81B[\uDFE0-\uDFFF]'
1160 },
1161 {
1162     name: 'InImperial_Aramaic',
1163     astral: '\uD802[\uDC40-\uDC5F]'
1164 },
1165 {
1166     name: 'InInscriptional_Pahlavi',
1167     astral: '\uD802[\uDF60-\uDF7F]'
1168 },
1169 {
1170     name: 'InInscriptional_Parthian',
1171     astral: '\uD802[\uDF40-\uDF5F]'
1172 },
1173 {
1174     name: 'InJavanese',
1175     bmp: '\uA980-\uA9DF'
1176 },
1177 {
1178     name: 'InKaithi',
1179     astral: '\uD804[\uDC80-\uDCCF]'
1180 },
1181 {
1182     name: 'InKana_Supplement',
1183     astral: '\uD82C[\uDC00-\uDCFF]'
1184 },
1185 {
1186     name: 'InKanbun',
1187     bmp: '\u3190-\u319F'
1188 },
1189 {
1190     name: 'InKangxi_Radicals',
1191     bmp: '\u2F00-\u2FDF'
1192 },
1193 {
1194     name: 'InKannada',
1195     bmp: '\u0C80-\u0CFF'
1196 },
1197 {
1198     name: 'InKatakana',
1199     bmp: '\u30A0-\u30FF'
1200 },
1201 {
1202     name: 'InKatakana_Phonetic_Extensions',
1203     bmp: '\u31F0-\u31FF'
1204 },
1205 {
1206     name: 'InKayah_Li',
1207     bmp: '\uA900-\uA92F'
1208 },
1209 {
1210     name: 'InKharoshthi',
1211     astral: '\uD802[\uDE00-\uDE5F]'
1212 },
1213 {
1214     name: 'InKhmer',
1215     bmp: '\u1780-\u17FF'
1216 },
1217 {
1218     name: 'InKhmer_Symbols',
1219     bmp: '\u19E0-\u19FF'
1220 },
1221 {
1222     name: 'InKhojki',
1223     astral: '\uD804[\uDE00-\uDE4F]'
1224 },
1225 {
1226     name: 'InKhudawadi',
1227     astral: '\uD804[\uDEB0-\uDEFF]'
1228 },
1229 {
1230     name: 'InLao',
1231     bmp: '\u0E80-\u0EFF'
1232 },
1233 {
1234     name: 'InLatin_Extended_Additional',
1235     bmp: '\u1E00-\u1EFF'
1236 },
1237 {

```

```

1238     name: 'InLatin_Extended_A',
1239     bmp: '\u0100-\u017F'
1240 },
1241 {
1242     name: 'InLatin_Extended_B',
1243     bmp: '\u0180-\u024F'
1244 },
1245 {
1246     name: 'InLatin_Extended_C',
1247     bmp: '\u2C60-\u2C7F'
1248 },
1249 {
1250     name: 'InLatin_Extended_D',
1251     bmp: '\uA720-\uA7FF'
1252 },
1253 {
1254     name: 'InLatin_Extended_E',
1255     bmp: '\uAB30-\uAB6F'
1256 },
1257 {
1258     name: 'InLatin_1_Supplement',
1259     bmp: '\x80-\xFF'
1260 },
1261 {
1262     name: 'InLepcha',
1263     bmp: '\u1C00-\u1C4F'
1264 },
1265 {
1266     name: 'InLetterlike_Symbols',
1267     bmp: '\u2100-\u214F'
1268 },
1269 {
1270     name: 'InLimbu',
1271     bmp: '\u1900-\u194F'
1272 },
1273 {
1274     name: 'InLinear_A',
1275     astral: '\uD801[\uDE00-\uDF7F]'
1276 },
1277 {
1278     name: 'InLinear_B_Ideograms',
1279     astral: '\uD800[\uDC80-\uDCFF]'
1280 },
1281 {
1282     name: 'InLinear_B_Syllabary',
1283     astral: '\uD800[\uDC00-\uDC7F]'
1284 },
1285 {
1286     name: 'InLisu',
1287     bmp: '\uA4D0-\uA4FF'
1288 },
1289 {
1290     name: 'InLow_Surrogates',
1291     bmp: '\uDC00-\uDFFF'
1292 },
1293 {
1294     name: 'InLycian',
1295     astral: '\uD800[\uDE80-\uDE9F]'
1296 },
1297 {
1298     name: 'InLydian',
1299     astral: '\uD802[\uDD20-\uDD3F]'
1300 },
1301 {
1302     name: 'InMahajani',
1303     astral: '\uD804[\uDD50-\uDD7F]'
1304 },
1305 {
1306     name: 'InMahjong_Tiles',
1307     astral: '\uD83C[\uDC00-\uDC2F]'
1308 },
1309 {
1310     name: 'InMalayalam',
1311     bmp: '\u0D00-\u0D7F'
1312 },
1313 {
1314     name: 'InMandaic',
1315     bmp: '\u0840-\u085F'
1316 },
1317 {
1318     name: 'InManichaean',
1319     astral: '\uD802[\uDEC0-\uDEFF]'
1320 },
1321 {
1322     name: 'InMarchen',
1323     astral: '\uD807[\uDC70-\uDCBF]'
1324 },
1325 {
1326     name: 'InMathematical_Alphanumeric_Symbols',
1327     astral: '\uD835[\uDC00-\uDFFF]'
1328 },
1329 {
1330     name: 'InMathematical_Operators',
1331     bmp: '\u2200-\u22FF'
1332 },
1333 {
1334     name: 'InMeetei_Mayek',
1335     bmp: '\uABC0-\uABFF'
1336 },
1337 {
1338     name: 'InMeetei_Mayek_Extensions',
1339     bmp: '\uAAE0-\uAAFF'
1340 },
1341 {
1342     name: 'InMende_Kikakui',
1343     astral: '\uD83A[\uDC00-\uDCDF]'
1344 },
1345 {
1346     name: 'InMeroitic_Cursive',
1347     astral: '\uD802[\uDDA0-\uDDFF]'
1348 },
1349 {
1350     name: 'InMeroitic_Hieroglyphs',
1351     astral: '\uD802[\uDD80-\uDD9F]'
1352 },

```

```

1353 {
1354     name: 'InMiao',
1355     astral: '\uD81B\uDF00-\uDF9F]'
1356 },
1357 {
1358     name: 'InMiscellaneous_Mathematical_Symbols_A',
1359     bmp: '\u27C0-\u27EF'
1360 },
1361 {
1362     name: 'InMiscellaneous_Mathematical_Symbols_B',
1363     bmp: '\u2980-\u29FF'
1364 },
1365 {
1366     name: 'InMiscellaneous_Symbols',
1367     bmp: '\u2600-\u26FF'
1368 },
1369 {
1370     name: 'InMiscellaneous_Symbols_and_Arrows',
1371     bmp: '\u2B00-\u2BFF'
1372 },
1373 {
1374     name: 'InMiscellaneous_Symbols_and_Pictographs',
1375     astral: '\uD83C[\uDF00-\uDFFF][\uD83D[\uDC00-\uDFFF]'
1376 },
1377 {
1378     name: 'InMiscellaneous_Technical',
1379     bmp: '\u2300-\u23FF'
1380 },
1381 {
1382     name: 'InModi',
1383     astral: '\uD805\uDE00-\uDE5F]'
1384 },
1385 {
1386     name: 'InModifier_Tone_Letters',
1387     bmp: '\uA700-\uA71F'
1388 },
1389 {
1390     name: 'InMongolian',
1391     bmp: '\u1800-\u18AF'
1392 },
1393 {
1394     name: 'InMongolian_Supplement',
1395     astral: '\uD805\uDE60-\uDE7F]'
1396 },
1397 {
1398     name: 'InMro',
1399     astral: '\uD81A\uDE40-\uDE6F]'
1400 },
1401 {
1402     name: 'InMultani',
1403     astral: '\uD804\uDE80-\uDEAF]'
1404 },
1405 {
1406     name: 'InMusical_Symbols',
1407     astral: '\uD834[\uDD00-\uDFFF]'
1408 },
1409 {
1410     name: 'InMyanmar',
1411     bmp: '\u1000-\u109F'
1412 },
1413 {
1414     name: 'InMyanmar_Extended_A',
1415     bmp: '\uAA60-\uAA7F'
1416 },
1417 {
1418     name: 'InMyanmar_Extended_B',
1419     bmp: '\uA9E0-\uA9FF'
1420 },
1421 {
1422     name: 'InNko',
1423     bmp: '\u07C0-\u07FF'
1424 },
1425 {
1426     name: 'InNabataean',
1427     astral: '\uD802[\uDC80-\uDCAF]'
1428 },
1429 {
1430     name: 'InNew_Tai_Lue',
1431     bmp: '\u1980-\u19DF'
1432 },
1433 {
1434     name: 'InNwa',
1435     astral: '\uD805[\uDC00-\uDC7F]'
1436 },
1437 {
1438     name: 'InNumber_Forms',
1439     bmp: '\u2150-\u218F'
1440 },
1441 {
1442     name: 'InOgham',
1443     bmp: '\u1680-\u169F'
1444 },
1445 {
1446     name: 'InOl_Chiki',
1447     bmp: '\u1C50-\u1C7F'
1448 },
1449 {
1450     name: 'InOld_Hungarian',
1451     astral: '\uD803[\uDC80-\uDCFF]'
1452 },
1453 {
1454     name: 'InOld_Italic',
1455     astral: '\uD800[\uDF00-\uDF2F]'
1456 },
1457 {
1458     name: 'InOld_North_Arabian',
1459     astral: '\uD802[\uDE80-\uDE9F]'
1460 },
1461 {
1462     name: 'InOld_Permic',
1463     astral: '\uD800[\uDF50-\uDF7F]'
1464 },
1465 {
1466     name: 'InOld_Persian',
1467     astral: '\uD800[\uDFA0-\uDFDF]'

```

```
1468 },
1469 {
1470   name: 'InOld_South_Arabian',
1471   astral: '\uD802\uDE60\uDE7F'
1472 },
1473 {
1474   name: 'InOld_Turkic',
1475   astral: '\uD803\uDC00\uDC4F'
1476 },
1477 {
1478   name: 'InOptical_Character_Recognition',
1479   bmp: '\u2440\u245F'
1480 },
1481 {
1482   name: 'InOriya',
1483   bmp: '\u0B00\u0B7F'
1484 },
1485 {
1486   name: 'InOrnamental_Dingbats',
1487   astral: '\uD83D\uDE50\uDE7F'
1488 },
1489 {
1490   name: 'InOsage',
1491   astral: '\uD801\uDCB0\uDCFF'
1492 },
1493 {
1494   name: 'InOsmanya',
1495   astral: '\uD801\uDC80\uDCAF'
1496 },
1497 {
1498   name: 'InPahawh_Hmong',
1499   astral: '\uD81A\uDF00\uDF8F'
1500 },
1501 {
1502   name: 'InPalmyrene',
1503   astral: '\uD802\uDC60\uDC7F'
1504 },
1505 {
1506   name: 'InPau_Cin_Hau',
1507   astral: '\uD806\uDEC0\uDEFF'
1508 },
1509 {
1510   name: 'InPhags_pa',
1511   bmp: '\uA840\uA87F'
1512 },
1513 {
1514   name: 'InPhaistos_Disc',
1515   astral: '\uD800\uDDDD\uDDFF'
1516 },
1517 {
1518   name: 'InPhoenician',
1519   astral: '\uD802\uDD00\uDD1F'
1520 },
1521 {
1522   name: 'InPhonetic_Extensions',
1523   bmp: '\u1D00\u1D7F'
1524 },
1525 {
1526   name: 'InPhonetic_Extensions_Supplement',
1527   bmp: '\u1D80\u1DBF'
1528 },
1529 {
1530   name: 'InPlaying_Cards',
1531   astral: '\uD83C\uDCA0\uDCFF'
1532 },
1533 {
1534   name: 'InPrivate_Use_Area',
1535   bmp: '\uE000\uF8FF'
1536 },
1537 {
1538   name: 'InPsalter_Pahlavi',
1539   astral: '\uD802\uDF80\uDFAF'
1540 },
1541 {
1542   name: 'InRejang',
1543   bmp: '\uA930\uA95F'
1544 },
1545 {
1546   name: 'InRumi_Numeral_Symbols',
1547   astral: '\uD803\uDE60\uDE7F'
1548 },
1549 {
1550   name: 'InRunic',
1551   bmp: '\u16A0\u16FF'
1552 },
1553 {
1554   name: 'InSamaritan',
1555   bmp: '\u0800\u083F'
1556 },
1557 {
1558   name: 'InSaurashtra',
1559   bmp: '\uA880\uA8DF'
1560 },
1561 {
1562   name: 'InSharada',
1563   astral: '\uD804\uDD80\uDDDF'
1564 },
1565 {
1566   name: 'InShavian',
1567   astral: '\uD801\uDC50\uDC7F'
1568 },
1569 {
1570   name: 'InShorthand_Format_Controls',
1571   astral: '\uD82F\uDCA0\uDCAF'
1572 },
1573 {
1574   name: 'InSiddham',
1575   astral: '\uD805\uDD80\uDDFF'
1576 },
1577 {
1578   name: 'InSinhala',
1579   bmp: '\u0D80\u0DFF'
1580 },
1581 {
1582   name: 'InSinhala_Archaic_Numbers',
```

```

1583     astral: '\uD804[\uDDE0-\uDDFF]'
1584 },
1585 {
1586     name: 'InSmall_Form_Variants',
1587     bmp: '\uFE50-\uFE6F'
1588 },
1589 {
1590     name: 'InSora_Sompeng',
1591     astral: '\uD804[\uDCD0-\uDCFF]'
1592 },
1593 {
1594     name: 'InSpacing_Modifier_Letters',
1595     bmp: '\u02B0-\u02FF'
1596 },
1597 {
1598     name: 'InSpecials',
1599     bmp: '\uFFFF-\uFFFF'
1600 },
1601 {
1602     name: 'InSundanese',
1603     bmp: '\u1B80-\u1BBF'
1604 },
1605 {
1606     name: 'InSundanese_Supplement',
1607     bmp: '\u1CC0-\u1CCF'
1608 },
1609 {
1610     name: 'InSuperscripts_and_Subscripts',
1611     bmp: '\u2070-\u209F'
1612 },
1613 {
1614     name: 'InSupplemental_Arrows_A',
1615     bmp: '\u27F0-\u27FF'
1616 },
1617 {
1618     name: 'InSupplemental_Arrows_B',
1619     bmp: '\u2900-\u297F'
1620 },
1621 {
1622     name: 'InSupplemental_Arrows_C',
1623     astral: '\uD83E[\uDC00-\uDCFF]'
1624 },
1625 {
1626     name: 'InSupplemental_Mathematical_Operators',
1627     bmp: '\u2A00-\u2AFF'
1628 },
1629 {
1630     name: 'InSupplemental_Punctuation',
1631     bmp: '\u2E00-\u2E7F'
1632 },
1633 {
1634     name: 'InSupplemental_Symbols_and_Pictographs',
1635     astral: '\uD83E[\uDD00-\uDDFF]'
1636 },
1637 {
1638     name: 'InSupplementary_Private_Use_Area_A',
1639     astral: ' [\uDB80-\uDBBF][\uDC00-\uDFFF]'
1640 },
1641 {
1642     name: 'InSupplementary_Private_Use_Area_B',
1643     astral: ' [\uDBC0-\uDBFF][\uDC00-\uDFFF]'
1644 },
1645 {
1646     name: 'InSutton_SignWriting',
1647     astral: '\uD836[\uDC00-\uDEAF]'
1648 },
1649 {
1650     name: 'InSyloti_Nagri',
1651     bmp: '\uA800-\uA82F'
1652 },
1653 {
1654     name: 'InSyriac',
1655     bmp: '\u0700-\u074F'
1656 },
1657 {
1658     name: 'InTagalog',
1659     bmp: '\u1700-\u171F'
1660 },
1661 {
1662     name: 'InTagbanwa',
1663     bmp: '\u1760-\u177F'
1664 },
1665 {
1666     name: 'InTags',
1667     astral: '\uDB40[\uDC00-\uDC7F]'
1668 },
1669 {
1670     name: 'InTai_Le',
1671     bmp: '\u1950-\u197F'
1672 },
1673 {
1674     name: 'InTai_Tham',
1675     bmp: '\u1A20-\u1AAF'
1676 },
1677 {
1678     name: 'InTai_Viet',
1679     bmp: '\uAA80-\uAADF'
1680 },
1681 {
1682     name: 'InTai_Xuan_Jing_Symbols',
1683     astral: '\uD834[\uDF00-\uDF5F]'
1684 },
1685 {
1686     name: 'InTakri',
1687     astral: '\uD805[\uDE80-\uDECF]'
1688 },
1689 {
1690     name: 'InTamil',
1691     bmp: '\u0B80-\u0BFF'
1692 },
1693 {
1694     name: 'InTangut',
1695     astral: ' [\uD81C-\uD821][\uDC00-\uDFFF]'
1696 },
1697 {

```

```

1698     name: 'InTangut_Components',
1699     astral: '\uD822[\uDC00-\uDEFF]'
1700 },
1701 {
1702     name: 'InTelugu',
1703     bmp: '\u0C00-\u0C7F'
1704 },
1705 {
1706     name: 'InThaana',
1707     bmp: '\u0780-\u07BF'
1708 },
1709 {
1710     name: 'InThai',
1711     bmp: '\u0E00-\u0E7F'
1712 },
1713 {
1714     name: 'InTibetan',
1715     bmp: '\u0F00-\u0FFF'
1716 },
1717 {
1718     name: 'InTifinagh',
1719     bmp: '\u2D30-\u2D7F'
1720 },
1721 {
1722     name: 'InTirhuta',
1723     astral: '\uD805[\uDC80-\uDCDF]'
1724 },
1725 {
1726     name: 'InTransport_and_Map_Symbols',
1727     astral: '\uD83D[\uDE80-\uDEFF]'
1728 },
1729 {
1730     name: 'InUgaritic',
1731     astral: '\uD800[\uDF80-\uDF9F]'
1732 },
1733 {
1734     name: 'InUnified_Canadian_Aboriginal_Syllabics',
1735     bmp: '\u1400-\u167F'
1736 },
1737 {
1738     name: 'InUnified_Canadian_Aboriginal_Syllabics_Extended',
1739     bmp: '\u18B0-\u18FF'
1740 },
1741 {
1742     name: 'InVai',
1743     bmp: '\uA500-\uA63F'
1744 },
1745 {
1746     name: 'InVariation_Selectors',
1747     bmp: '\uFE00-\uFE0F'
1748 },
1749 {
1750     name: 'InVariation_Selectors_Supplement',
1751     astral: '\uDB40[\uDD00-\uDDDF]'
1752 },
1753 {
1754     name: 'InVedic_Extensions',
1755     bmp: '\u1CD0-\u1CFF'
1756 },
1757 {
1758     name: 'InVertical_Forms',
1759     bmp: '\uFE10-\uFE1F'
1760 },
1761 {
1762     name: 'InWarang_Citi',
1763     astral: '\uD806[\uDCA0-\uDCFF]'
1764 },
1765 {
1766     name: 'InYi_Radicals',
1767     bmp: '\uA490-\uA4CF'
1768 },
1769 {
1770     name: 'InYi_Syllables',
1771     bmp: '\uA000-\uA48F'
1772 },
1773 {
1774     name: 'InYijing_Hexagram_Symbols',
1775     bmp: '\u4DC0-\u4DFF'
1776 }
1777 ];
1778
1779 };
1780
1781 },{}],5:(function(require,module,exports){
1782 /*!
1783  * XRegExp Unicode Categories 3.2.0
1784  * <xregexp.com>
1785  * Steven Levithan (c) 2010-2017 MIT License
1786  * Unicode data by Mathias Bynens <mathiasbynens.be>
1787  */
1788
1789 module.exports = function(XRegExp) {
1790     'use strict';
1791
1792     /**
1793      * Adds support for Unicode's general categories. E.g., `p{Lu}` or `p{Uppercase Letter}`. See
1794      * category descriptions in UAX #44 <http://unicode.org/reports/tr44/#GC_Values_Table>. Token
1795      * names are case insensitive, and any spaces, hyphens, and underscores are ignored.
1796      *
1797      * Uses Unicode 9.0.0.
1798      *
1799      * @requires XRegExp, Unicode Base
1800      */
1801
1802     if (!XRegExp.addUnicodeData) {
1803         throw new ReferenceError('Unicode Base must be loaded before Unicode Categories');
1804     }
1805
1806     XRegExp.addUnicodeData([
1807         {
1808             name: 'C',
1809             alias: 'Other',
1810             isBmpLast: true,
1811             bmp: '\0-\x1F\x7F-\x9F\xAD\u0378\u0379\u0380-\u0383\u038B\u038D\u03A2\u0530\u0557\u0558\u0560\u0588\u058B\u058C\u0590\u05C8-\u05CF\u05EB-\u05EF\u05F5-\u0605\u061C\u
1812             astral: '\uD800[\uDC0C\uDC27\uDC3B\uDC3E\uDC4E\uDC4F\uDC5E-\uDC7F\uDCFB\uDCFF\uDD03-\uDD06\uDD34-\uDD36\uDD8F\uDD9C-\uDD9F\uDDA1-\uDDCF\uDDFE-\uDE7F\uDE9D-\u

```



```
1813 },
1814 {
1815     name: 'Cc',
1816     alias: 'Control',
1817     bmp: '\0-\x1F\x7F-\xF'
1818 },
1819 {
1820     name: 'Cf',
1821     alias: 'Format',
1822     bmp: '\xAD\u0600-\u0605\u061C\u06DD\u070F\u08E2\u180E\u200B-\u200F\u202A-\u202E\u2060-\u2064\u2066-\u206F\uFEFF\uFFFF-\uFFFF',
1823     astral: '\uD804\uDCBD\uDB2F\uDCA0-\uDCA3][\uD834[\uD873-\uDD7A][\uDB40[\uDC01\uDC20-\uDC7F]
1824 },
1825 {
1826     name: 'Cn',
1827     alias: 'Unassigned',
1828     bmp: '\u0378\u0379\u0380-\u0383\u038B\u038D\u03A2\u0530\u0557\u0558\u0560\u0588\u058B\u058C\u0590\u05C8-\u05CF\u05EB-\u05EF\u05F5-\u05FF\u061D\u070E\u074B\u074C\u
1829     astral: '\uD800[\uDC0C\uDC27\uDC3B\uDC3E\uDC4E\uDC4F\uDC5E-\uDC7F\uDCFB-\uDCFF\uDD03-\uDD06\uDD34-\uDD36\uDD8F\uDD9C-\uDD9F\uDDA1-\uDDCF\uDDFE-\uDE7F\uDE9D-\u
1830 },
1831 {
1832     name: 'Co',
1833     alias: 'Private_Use',
1834     bmp: '\uE000-\uF8FF',
1835     astral: '[\uDB80-\uDBBE\uDBC0-\uDBFE][\uDC00-\uDFFF][\uDBBF\uDBFF][\uDC00-\uDFFD]'
1836 },
1837 {
1838     name: 'Cs',
1839     alias: 'Surrogate',
1840     bmp: '\uD800-\uDFFF'
1841 },
1842 {
1843     name: 'L',
1844     alias: 'Letter',
1845     bmp: 'A-Za-z\u00AA\u00B5\u00BA\u00C0-\u00D6\u00D8-\u00F6\u00F8-\u02C1\u02C6-\u02D1\u02E0-\u02E4\u02EC\u02EE\u0370-\u0374\u0376\u0377\u037A-\u037D\u037F\u0386\u0388-\u038A\u038C\u03
1846     astral: '\uD800[\uDC00-\uDC0B\uDC0D-\uDC26\uDC28-\uDC3A\uDC3C\uDC3D\uDC3F-\uDC4D\uDC50-\uDC5D\uDC80-\uDCFA\uDE80-\uDE9C\uDEA0-\uDED0\uDF00-\uDF1F\uDF30-\uDF4C
1847 },
1848 {
1849     name: 'Ll',
1850     alias: 'Lowercase_Letter',
1851     bmp: 'a-z\u00B5\u00DF-\u00F6\u00F8-\u02C1\u02C6-\u02D1\u02E0-\u02E4\u02EC\u02EE\u0374\u037A\u0559\u0640\u06E5\u06E6\u07F4\u07F5\u07FA\u081A\u0824\u0828\u0971\u0E46\u0EC6\u10FC\u17D7\u
1852     astral: '\uD801[\uDC28-\uDC4F\uDCD8-\uDCF8][\uD803[\uDC00-\uDCF2][\uD806[\uDC00-\uDCF7][\uD835[\uDC1A-\uDC33\uDC4E-\uDC54\uDC56-\uDC67\uDC82-\uDC9B\uDCB6-\uDCB9\u
1853 },
1854 {
1855     name: 'Lm',
1856     alias: 'Modifier_Letter',
1857     bmp: '\u02B0-\u02C1\u02C6-\u02D1\u02E0-\u02E4\u02EC\u02EE\u0374\u037A\u0559\u0640\u06E5\u06E6\u07F4\u07F5\u07FA\u081A\u0824\u0828\u0971\u0E46\u0EC6\u10FC\u17D7\u
1858     astral: '\uD81A[\uDF40-\uDF43][\uD81B[\uDF93-\uDF9F\uDFE0]'
1859 },
1860 {
1861     name: 'Lo',
1862     alias: 'Other_Letter',
1863     bmp: '\u00AA\u00BA\u01BB\u01C0-\u01C3\u0294\u05D0-\u05EA\u05F0-\u05F2\u0620-\u063F\u0641-\u064A\u066E\u066F\u0671-\u06D3\u06D5\u06EE\u06EF\u06FA-\u06FC\u06FF\u0710\u07
1864     astral: '\uD800[\uDC00-\uDC0B\uDC0D-\uDC26\uDC28-\uDC3A\uDC3C\uDC3D\uDC3F-\uDC4D\uDC50-\uDC5D\uDC80-\uDCFA\uDE80-\uDE9C\uDEA0-\uDED0\uDF00-\uDF1F\uDF30-\uDF4C
1865 },
1866 {
1867     name: 'Lt',
1868     alias: 'Titlecase_Letter',
1869     bmp: '\u01C5\u01C8\u01CB\u01F2\u1F88-\u1F8F\u1F98-\u1F9F\u1FA8-\u1FAF\u1FBC\u1FCC\u1FFC'
1870 },
1871 {
1872     name: 'Lu',
1873     alias: 'Uppercase_Letter',
1874     bmp: 'A-Z\u00C0-\u00D6\u00D8-\u00DE\u0100\u0102\u0104\u0106\u0108\u010A\u010C\u010E\u0110\u0112\u0114\u0116\u0118\u011A\u011C\u011E\u0120\u0122\u0124\u0126\u0128\u012A\u01
1875     astral: '\uD801[\uDC00-\uDC27\uDCB0-\uDCD3][\uD803[\uDC80-\uDCB2][\uD806[\uDCA0-\uDCBF][\uD835[\uDC00-\uDC19\uDC34-\uDC4D\uDC68-\uDC81\uDC9C\uDC9E\uDC9F\uDCA2\u
1876 },
1877 {
1878     name: 'M',
1879     alias: 'Mark',
1880     bmp: '\u0300-\u036F\u0483-\u0489\u0591-\u05BD\u05BF\u05C1\u05C2\u05C4\u05C5\u05C7\u0610-\u061A\u064B-\u065F\u0670\u06D6-\u06DC\u06DF-\u06E4\u06E7\u06E8\u06EA-\u06
1881     astral: '\uD800[\uDDFD\uDEE0\uDF76-\uDF7A][\uD802[\uDE01-\uDE03\uDE05\uDE06\uDE0C-\uDE0F\uDE38-\uDE3A\uDE3F\uDEE5\uDEE6][\uD804[\uDC00-\uDC02\uDC38-\uDC46\uDC7F-\uDC81]
1882 },
1883 {
1884     name: 'Mc',
1885     alias: 'Spacing_Mark',
1886     bmp: '\u0903\u093B\u093E-\u0940\u0949-\u094C\u094E\u094F\u0982\u0983\u098E-\u09C0\u09C7\u09C8\u09CB\u09CC\u09D7\u0A03\u0A3E-\u0A40\u0A83\u0ABE-\u0AC0\u0AC9\u0AC
1887     astral: '\uD804[\uDC00\uDC02\uDC82\uDCB0-\uDCB2\uDCB7\uDCB8\uDD2C\uDD82\uDD83\uDD85\uDD8F\uDDC0\uDE2C-\uDE2E\uDE32\uDE33\uDE35\uDEE0-\uDEE2\uDF02\uDF03\uDF
1888 },
1889 {
1890     name: 'Me',
1891     alias: 'Enclosing_Mark',
1892     bmp: '\u0488\u0489\u1ABE\u20DD-\u20E0\u20E2-\u20E4\uA670-\uA672'
1893 },
1894 {
1895     name: 'Mn',
1896     alias: 'Nonspacing_Mark',
1897     bmp: '\u0300-\u036F\u0483-\u0487\u0591-\u05BD\u05BF\u05C1\u05C2\u05C4\u05C5\u05C7\u0610-\u061A\u064B-\u065F\u0670\u06D6-\u06DC\u06DF-\u06E4\u06E7\u06E8\u06EA-\u06
1898     astral: '\uD800[\uDDFD\uDEE0\uDF76-\uDF7A][\uD802[\uDE01-\uDE03\uDE05\uDE06\uDE0C-\uDE0F\uDE38-\uDE3A\uDE3F\uDEE5\uDEE6][\uD804[\uDC01\uDC38-\uDC46\uDC7F-\uDC81]
1899 },
1900 {
1901     name: 'N',
1902     alias: 'Number',
1903     bmp: '0-9\u00B2\u00B3\u00B9\u00BC-\u00BE\u0660-\u0669\u06F0-\u06F9\u07C0-\u07C9\u0966-\u096F\u09E6-\u09EF\u09F4-\u09F9\u0A66-\u0A6F\u0AE6-\u0AEF\u0B66-\u0B6F\u0B72-\u0B77\u0BE
1904     astral: '\uD800[\uDD07\uDD33\uDD40-\uDD78\uDD8A\uDD8B\uDEE1-\uDEFB\uDF20-\uDF23\uDF41\uDF4A\uDFD1-\uDFD5][\uD801[\uDCA0-\uDCA9][\uD802[\uDC58-\uDC5F\uDC79-\uDC
1905 },
1906 {
1907     name: 'Nd',
1908     alias: 'Decimal_Number',
1909     bmp: '0-9\u0660-\u0669\u06F0-\u06F9\u07C0-\u07C9\u0966-\u096F\u09E6-\u09EF\u0A66-\u0A6F\u0AE6-\u0AEF\u0B66-\u0B6F\u0BE6-\u0BEF\u0C66-\u0C6F\u0CE6-\u0CEF\u0D66-\u0D6
1910     astral: '\uD801[\uDCA0-\uDCA9][\uD804[\uDC66-\uDC6F\uDCF0-\uDCF9\uDD36-\uDD3F\uDDD0-\uDDD9\uDEF0-\uDEF9][\uD805[\uDC50-\uDC59\uDCD0-\uDCD9\uDE50-\uDE59\uDEC0-\uE
1911 },
1912 {
1913     name: 'Ni',
1914     alias: 'Letter_Number',
1915     bmp: '\u16EE-\u16F0\u2160-\u2182\u2185-\u2188\u3007\u3021-\u3029\u3038-\u303A\uA6E6-\uA6EF',
1916     astral: '\uD800[\uDD40-\uDD74\uDF41\uDF4A\uDFD1-\uDFD5][\uD809[\uDC00-\uDC6E]'
1917 },
1918 {
1919     name: 'No',
1920     alias: 'Other_Number',
1921     bmp: '\u00B2\u00B3\u00B9\u00BC-\u00BE\u09F4-\u09F9\u0B72-\u0B77\u0BF0-\u0BF2\u0C78-\u0C7E\u0D58-\u0D5E\u0D70-\u0D78\u0F2A-\u0F33\u1369-\u137C\u17F0-\u17F9\u19DA\u2070\u2074-
1922     astral: '\uD800[\uDD07\uDD33\uDD75-\uDD78\uDD8A\uDD8B\uDEE1-\uDEFB\uDF20-\uDF23][\uD802[\uDC58-\uDC5F\uDC79-\uDC7F\uDCA7-\uDCAF\uDCFB-\uDCFF\uDD16-\uDD1B\uDDE
1923 },
1924 {
1925     name: 'P',
1926     alias: 'Punctuation',
1927     bmp: '\u0021-\u0023\u0025-\u002A\u002C-\u002F\u003A\u003B\u003F\u0040\u005B-\u005D\u005F\u007B\u007D\u00A1\u00A7\u00AB\u00B6\u00B7\u00BB\u00BF\u0037\u0055A-\u0055F\u00589\u0058A\u0058E\u005C\u005C3\u005C6\u00!
```

```
1928     astral: '\uD800\uDD00\uDD02\uDF9F\uDFFD0]]\uD801\uDD6F\uD802\uDC57\uDD1F\uDD3F\uDE50-\uDE58\uDE7F\uDEF0-\uDEF6\uDF39-\uDF3F\uDF99-\uDF9C]]\uD804\uDC47-\uDC4D
1929     },
1930     {
1931         name: 'Pc',
1932         alias: 'Connector_Punctuation',
1933         bmp: '\x5F\u203F\u2040\u2054\uFE33\uFE34\uFE4D-\uFE4F\uFF3F'
1934     },
1935     {
1936         name: 'Pd',
1937         alias: 'Dash_Punctuation',
1938         bmp: '\x2D\u058A\u05BE\u1400\u1806\u2010-\u2015\u2E17\u2E1A\u2E3A\u2E3B\u2E40\u301C\u3030\u30A0\uFE31\uFE32\uFE58\uFE63\uFF0D'
1939     },
1940     {
1941         name: 'Pe',
1942         alias: 'Close_Punctuation',
1943         bmp: '\x29\x5D\x7D\u0F3B\u0F3D\u169C\u2046\u207E\u208E\u2309\u230B\u232A\u2769\u276B\u276D\u276F\u2771\u2773\u2775\u27C6\u27E7\u27E9\u27EB\u27ED\u27EF\u2984\u2
1944     },
1945     {
1946         name: 'PF',
1947         alias: 'Final_Punctuation',
1948         bmp: '\xB8\u2019\u201D\u203A\u2E03\u2E05\u2E0A\u2E0D\u2E1D\u2E21'
1949     },
1950     {
1951         name: 'Pi',
1952         alias: 'Initial_Punctuation',
1953         bmp: '\xAB\u2018\u201B\u201C\u201F\u2039\u2E02\u2E04\u2E09\u2E0C\u2E1C\u2E20'
1954     },
1955     {
1956         name: 'Po',
1957         alias: 'Other_Punctuation',
1958         bmp: '\x21-\x23\x25-\x27\\x2A\x2C\\x2E\x2F\x3A\x3B\\x3F\x40\\x5C\xA1\xA7\xB6\xB7\xBF\u037E\u0387\u055A-\u055F\u0589\u05C0\u05C3\u05C6\u05F3\u05F4\u0609\u060A\u060C\u0
1959         astral: '\uD800\uDD00\uDD02\uDF9F\uDFFD0]]\uD801\uDD6F\uD802\uDC57\uDD1F\uDD3F\uDE50-\uDE58\uDE7F\uDEF0-\uDEF6\uDF39-\uDF3F\uDF99-\uDF9C]]\uD804\uDC47-\uDC4D
1960     },
1961     {
1962         name: 'Ps',
1963         alias: 'Open_Punctuation',
1964         bmp: '\x28\x5B\x7B\u0F3A\u0F3C\u169B\u201A\u201E\u2045\u207D\u208D\u2308\u230A\u2329\u2768\u276A\u276C\u276E\u2770\u2772\u2774\u27C5\u27E6\u27E8\u27EA\u27EC\u
1965     },
1966     {
1967         name: 'S',
1968         alias: 'Symbol',
1969         bmp: '\x24\x2B\x3C-\x3E\x5E\x60\\x7C\x7E\xA2-\xA6\xA8\xA9\xAC\xAE-\xB1\xB4\xB8\xD7\xF7\u02C2-\u02C5\u02D2-\u02DF\u02E5-\u02EB\u02ED\u02EF\u02FF\u0375\u0384\u0385\u
1970         astral: '\uD800\uDD37\uDD3F\uDD79-\uDD89\uDD8C-\uDD8E\uDD90-\uDD9B\uDDA0\uDDD0\uDDFC]]\uD802\uDC77\uDC78\uDEC8]]\uD805\uDF3F\uD81A[\uDF3C-\uDF3F\uDF45]]\uD
1971     },
1972     {
1973         name: 'Sc',
1974         alias: 'Currency_Symbol',
1975         bmp: '\x24xA2-\xA5\u058F\u060B\u09F2\u09F3\u09FB\u0AF1\u0BF9\u0E3F\u17DB\u20A0-\u20BE\uA838\uFDFC\uFE69\uFF04\uFFE0\uFFE1\uFFE5\uFFE6'
1976     },
1977     {
1978         name: 'Sk',
1979         alias: 'Modifier_Symbol',
1980         bmp: '\x5E\x60\xA8\xAF\xB4\xB8\u02C2-\u02C5\u02D2-\u02DF\u02E5-\u02EB\u02ED\u02EF\u02FF\u0375\u0384\u0385\u1FBD\u1FBF\u1FC1\u1FCD\u1FCF\u1FDD-\u1FDF\u1FED-\u1F
1981         astral: '\uD83C[\uDFFB-\uDFFF]'
1982     },
1983     {
1984         name: 'Sm',
1985         alias: 'Math_Symbol',
1986         bmp: '\x2B\x3C-\x3E\x7C\x7E\xAC\xB1\xD7\xF7\u03F6\u0606-\u0608\u2044\u2052\u207A-\u207C\u208A-\u208C\u2118\u2140\u2144\u214B\u2190\u2194\u219A\u219B\u21A0\u21A
1987         astral: '\uD835[\uDECI\uDEDB\uDEFB\uDF15\uDF35\uDF4F\uDF6F\uDF89\uDFA9\uDFC3]]\uD83B[\uDEF0\uDEF1]'
1988     },
1989     {
1990         name: 'So',
1991         alias: 'Other_Symbol',
1992         bmp: '\xA6\xA9\xAE\xB0\u0482\u058D\u058E\u060F\u06DE\u06E9\u06FD\u06FE\u07F6\u09FA\u0B70\u0BF3-\u0BF8\u0BFA\u0C7F\u0D4F\u0D79\u0F01\u0F03\u0F13\u0F15\u0F
1993         astral: '\uD800\uDD37\uDD3F\uDD79-\uDD89\uDD8C-\uDD8E\uDD90-\uDD9B\uDDA0\uDDDD\uDDFC]]\uD802\uDC77\uDC78\uDEC8]]\uD805\uDF3F\uD81A[\uDF3C-\uDF3F\uDF45]]\uD
1994     },
1995     {
1996         name: 'Z',
1997         alias: 'Separator',
1998         bmp: '\x20xA0\u1680\u2000-\u200A\u2028\u2029\u202F\u205F\u3000'
1999     },
2000     {
2001         name: 'Zl',
2002         alias: 'Line_Separator',
2003         bmp: '\u2028'
2004     },
2005     {
2006         name: 'Zp',
2007         alias: 'Paragraph_Separator',
2008         bmp: '\u2029'
2009     },
2010     {
2011         name: 'Zs',
2012         alias: 'Space_Separator',
2013         bmp: '\x20xA0\u1680\u2000-\u200A\u202F\u205F\u3000'
2014     }
2015 ];
2016
2017 };
2018
2019 } , {} } , 6: { function (require) { module exports } {
2020 /*!
2021  * XRegExp Unicode Properties 3.2.0
2022  * <xregexp.com>
2023  * Steven Levithan (c) 2012-2017 MIT License
2024  * Unicode data by Mathias Bynens <mathiasbynens.be>
2025  */
2026
2027 module exports = function (XRegExp) {
2028     'use strict';
2029
2030     /**
2031      * Adds properties to meet the UTS #18 Level 1 RL1.2 requirements for Unicode regex support. See
2032      * <http://unicode.org/reports/tr18/#RL1.2>. Following are definitions of these properties from
2033      * UAX #44 <http://unicode.org/reports/tr44/>:
2034      *
2035      * - Alphabetic
2036      * Characters with the Alphabetic property. Generated from: Lowercase + Uppercase + Lt + Lm +
2037      * Lo + NI + Other_Alphabetic.
2038      *
2039      * - Default_Ignorable_Code_Point
2040      * For programmatic determination of default ignorable code points. New characters that should
2041      * be ignored in rendering (unless explicitly supported) will be assigned in these ranges,
2042      * permitting programs to correctly handle the default rendering of such characters when not
```

```

2043 * otherwise supported.
2044 *
2045 * - Lowercase
2046 * Characters with the Lowercase property. Generated from: Ll + Other_Lowercase.
2047 *
2048 * - Noncharacter_Code_Point
2049 * Code points permanently reserved for internal use.
2050 *
2051 * - Uppercase
2052 * Characters with the Uppercase property. Generated from: Lu + Other_Uppercase.
2053 *
2054 * - White_Space
2055 * Spaces, separator characters and other control characters which should be treated by
2056 * programming languages as "white space" for the purpose of parsing elements.
2057 *
2058 * The properties ASCII, Any, and Assigned are also included but are not defined in UAX #44. UTS
2059 * #18 RL1.2 additionally requires support for Unicode scripts and general categories. These are
2060 * included in XRegExp's Unicode Categories and Unicode Scripts addons.
2061 *
2062 * Token names are case insensitive, and any spaces, hyphens, and underscores are ignored.
2063 *
2064 * Uses Unicode 9.0.0.
2065 *
2066 * @requires XRegExp, Unicode Base
2067 */
2068
2069 if (!XRegExp.addUnicodeData) {
2070   throw new ReferenceError('Unicode Base must be loaded before Unicode Properties');
2071 }
2072
2073 var unicodeData = [
2074   {
2075     name: 'ASCII',
2076     bmp: '\0-\x7F'
2077   },
2078   {
2079     name: 'Alphabetic',
2080     bmp: 'A-Za-z\u00AA\u00B5\u00BA\u00C0-\u00D6\u00D8-\u00F6\u00F8-\u02C1\u02C6-\u02D1\u02E0-\u02E4\u02EC\u02EE\u0345\u0370-\u0374\u0376\u0377\u037A-\u037D\u037F\u0386\u0388-\u038A\u03
2081     astral: '\u0800[\uDC00-\uDC0B\uDC0D-\uDC26\uDC28-\uDC3A\uDC3C\uDC3D\uDC3F-\uDC4D\uDC50-\uDC5D\uDC80-\uDCFA\uDD40-\uDD74\uDE80-\uDE9C\uDEA0-\uDED0\uDF00-\uDF1
2082   },
2083   {
2084     name: 'Any',
2085     isBmpLast: true,
2086     bmp: '\0-\uFFFF',
2087     astral: '[\u0800-\uDBFF][\uDC00-\uDFFF]'
2088   },
2089   {
2090     name: 'Default_Ignorable_Code_Point',
2091     bmp: '\xAD\u034F\u061C\u115F\u1160\u17B4\u17B5\u180B\u180E\u200B\u200F\u202A-\u202E\u2060\u206F\u3164\uFE00-\uFE0F\uFEFF\uFFA0\uFFFF-\uFFFF',
2092     astral: '\u082F[\u0CA0-\u0CA3][\u0834[\u0D73-\u0D7A][\u0DB40-\u0DB43][\u0DC00-\u0DCFF]'
2093   },
2094   {
2095     name: 'Lowercase',
2096     bmp: 'a-z\u00AA\u00B5\u00BA\u00C0-\u00D6\u00D8-\u00F6\u00F8-\u00FF\u0101\u0103\u0105\u0107\u0109\u010B\u010D\u010F\u0111\u0113\u0115\u0117\u0119\u011B\u011D\u011F\u0121\u0123\u0125\u0127\u01
2097     astral: '\u0801[\uDC28-\uDC4F\uDCD8-\uDCFB][\u0803[\u0DCC0-\u0DCF2][\u0806[\u0DCC0-\u0DCF7][\u0835[\u0DC1A-\u0DC33\u0DC4E-\u0DC54\u0DC56-\u0DC67\u0DC82-\u0DC9B\u0DCB6-\u0DCB9\u
2098   },
2099   {
2100     name: 'Noncharacter_Code_Point',
2101     bmp: '\uFDD0-\uFDEF\uFFFE\uFFFF',
2102     astral: '[\u083F\u087F\u08BF\u08FF\u093F\u097F\u09BF\u09FF\uDA3F\uDA7F\uDABF\uDAFF\uDB3F\uDB7F\uDBBF\uDBFF][\uDFFE\uDFFF]'
2103   },
2104   {
2105     name: 'Uppercase',
2106     bmp: 'A-Z\u00C0-\u00D6\u00D8-\u00DE\u0100\u0102\u0104\u0106\u0108\u010A\u010C\u010E\u0110\u0112\u0114\u0116\u0118\u011A\u011C\u011E\u0120\u0122\u0124\u0126\u0128\u012A\u01
2107     astral: '\u0801[\uDC00-\uDC27\uDCB0-\uDCD3][\u0803[\uDC80-\uDCB2][\u0806[\u0DCA0-\u0DCBF][\u0835[\u0DC00-\u0DC19\u0DC34-\u0DC4D\u0DC68-\u0DC81\u0DC9C\u0DC9E\u0DC9F\u0DCA2\u
2108   },
2109   {
2110     name: 'White_Space',
2111     bmp: '\x09-\x0D\u20\u2000-\u200A\u2028\u2029\u202F\u205F\u3000'
2112   }
2113 ];
2114
2115 // Add non-generated data
2116 unicodeData.push({
2117   name: 'Assigned',
2118   // Since this is defined as the inverse of Unicode category Cn (Unassigned), the Unicode
2119   // Categories addon is required to use this property
2120   inverseOf: 'Cn'
2121 });
2122
2123 XRegExp.addUnicodeData(unicodeData);
2124
2125 };
2126
2127 },{}),7:(function(require,module,exports){
2128 /*!
2129 * XRegExp Unicode Scripts 3.2.0
2130 * <xregexp.com>
2131 * Steven Levithan (c) 2010-2017 MIT License
2132 * Unicode data by Mathias Bynens <mathiasbynens.be>
2133 */
2134
2135 module.exports = function(XRegExp) {
2136   'use strict';
2137
2138   /**
2139    * Adds support for all Unicode scripts. E.g., `p{Latin}`. Token names are case insensitive,
2140    * and any spaces, hyphens, and underscores are ignored.
2141    *
2142    * Uses Unicode 9.0.0.
2143    *
2144    * @requires XRegExp, Unicode Base
2145    */
2146
2147   if (!XRegExp.addUnicodeData) {
2148     throw new ReferenceError('Unicode Base must be loaded before Unicode Scripts');
2149   }
2150
2151   XRegExp.addUnicodeData([
2152     {
2153       name: 'Adlam',
2154       astral: '\u083A[\uDD00-\uDD4A\uDD50-\uDD59\uDD5E\uDD5F]'
2155     },
2156     {
2157       name: 'Ahom',

```

```

2158 astral: '\uD805[\uDF00-\uDF19\uDF1A-\uDF2B\uDF30-\uDF3F]'
2159 },
2160 {
2161     name: 'Anatolian_Hieroglyphs',
2162     astral: '\uD811[\uDC00-\uDE46]'
2163 },
2164 {
2165     name: 'Arabic',
2166     bmp: '\u0600-\u0604\u0606-\u060B\u060D-\u061A\u061E\u0620-\u063F\u0641-\u064A\u0656-\u066F\u0671-\u06DC\u06DE-\u06FF\u0750-\u077F\u08A0-\u08B4\u08B6-\u08BD\u08D4-\u08E0-\u08E3\u08E5-\u08E8\u08EA\u08EC\u08ED\u08F0\u08F4\u08F6\u08F8\u08FA\u08FD\u08FE\u0900-\u0903\u0905-\u0908\u090A\u090C\u090F\u0910\u0912\u0913\u0914\u0916\u0917\u0919\u091B\u091C\u091E\u091F\u0921\u0922\u0923\u0924\u0925\u0926\u0927\u0928\u0929\u092A\u092B\u092C\u092D\u092E\u092F\u0930\u0931\u0932\u0933\u0934\u0935\u0936\u0937\u0938\u0939\u093A\u093B\u093C\u093D\u093E\u093F\u0940\u0941\u0942\u0943\u0944\u0945\u0946\u0947\u0948\u0949\u094A\u094B\u094C\u094D\u094E\u094F\u0950\u0951\u0952\u0953\u0954\u0955\u0956\u0957\u0958\u0959\u095A\u095B\u095C\u095D\u095E\u095F\u0960\u0961\u0962\u0963\u0964\u0965\u0966\u0967\u0968\u0969\u096A\u096B\u096C\u096D\u096E\u096F\u0970\u0971\u0972\u0973\u0974\u0975\u0976\u0977\u0978\u0979\u097A\u097B\u097C\u097D\u097E\u097F\u0980\u0981\u0982\u0983\u0984\u0985\u0986\u0987\u0988\u0989\u098A\u098B\u098C\u098D\u098E\u098F\u0990\u0991\u0992\u0993\u0994\u0995\u0996\u0997\u0998\u0999\u099A\u099B\u099C\u099D\u099E\u099F\u09A0\u09A1\u09A2\u09A3\u09A4\u09A5\u09A6\u09A7\u09A8\u09A9\u09AA\u09AB\u09AC\u09AD\u09AE\u09AF\u09B0\u09B1\u09B2\u09B3\u09B4\u09B5\u09B6\u09B7\u09B8\u09B9\u09BA\u09BB\u09BC\u09BD\u09BE\u09BF\u09C0\u09C1\u09C2\u09C3\u09C4\u09C5\u09C6\u09C7\u09C8\u09C9\u09CA\u09CB\u09CC\u09CD\u09CE\u09CF\u09D0\u09D1\u09D2\u09D3\u09D4\u09D5\u09D6\u09D7\u09D8\u09D9\u09DA\u09DB\u09DC\u09DD\u09DE\u09DF\u09E0\u09E1\u09E2\u09E3\u09E4\u09E5\u09E6\u09E7\u09E8\u09E9\u09EA\u09EB\u09EC\u09ED\u09EE\u09EF\u09F0\u09F1\u09F2\u09F3\u09F4\u09F5\u09F6\u09F7\u09F8\u09F9\u09FA\u09FB\u09FC\u09FD\u09FE\u09FF\u0A00\u0A01\u0A02\u0A03\u0A04\u0A05\u0A06\u0A07\u0A08\u0A09\u0A0A\u0A0B\u0A0C\u0A0D\u0A0E\u0A0F\u0A10\u0A11\u0A12\u0A13\u0A14\u0A15\u0A16\u0A17\u0A18\u0A19\u0A1A\u0A1B\u0A1C\u0A1D\u0A1E\u0A1F\u0A20\u0A21\u0A22\u0A23\u0A24\u0A25\u0A26\u0A27\u0A28\u0A29\u0A2A\u0A2B\u0A2C\u0A2D\u0A2E\u0A2F\u0A30\u0A31\u0A32\u0A33\u0A34\u0A35\u0A36\u0A37\u0A38\u0A39\u0A3A\u0A3B\u0A3C\u0A3D\u0A3E\u0A3F\u0A40\u0A41\u0A42\u0A43\u0A44\u0A45\u0A46\u0A47\u0A48\u0A49\u0A4A\u0A4B\u0A4C\u0A4D\u0A4E\u0A4F\u0A50\u0A51\u0A52\u0A53\u0A54\u0A55\u0A56\u0A57\u0A58\u0A59\u0A5A\u0A5B\u0A5C\u0A5D\u0A5E\u0A5F\u0A60\u0A61\u0A62\u0A63\u0A64\u0A65\u0A66\u0A67\u0A68\u0A69\u0A6A\u0A6B\u0A6C\u0A6D\u0A6E\u0A6F\u0A70\u0A71\u0A72\u0A73\u0A74\u0A75\u0A76\u0A77\u0A78\u0A79\u0A7A\u0A7B\u0A7C\u0A7D\u0A7E\u0A7F\u0A80\u0A81\u0A82\u0A83\u0A84\u0A85\u0A86\u0A87\u0A88\u0A89\u0A8A\u0A8B\u0A8C\u0A8D\u0A8E\u0A8F\u0A90\u0A91\u0A92\u0A93\u0A94\u0A95\u0A96\u0A97\u0A98\u0A99\u0A9A\u0A9B\u0A9C\u0A9D\u0A9E\u0A9F\u0AA0\u0AA1\u0AA2\u0AA3\u0AA4\u0AA5\u0AA6\u0AA7\u0AA8\u0AA9\u0AAA\u0AAB\u0AAC\u0AAD\u0AAE\u0AAF\u0AB0\u0AB1\u0AB2\u0AB3\u0AB4\u0AB5\u0AB6\u0AB7\u0AB8\u0AB9\u0ABA\u0ABB\u0ABC\u0ABD\u0ABE\u0ABF\u0AC0\u0AC1\u0AC2\u0AC3\u0AC4\u0AC5\u0AC6\u0AC7\u0AC8\u0AC9\u0ACA\u0ACB\u0ACC\u0ACD\u0ACE\u0ACF\u0AD0\u0AD1\u0AD2\u0AD3\u0AD4\u0AD5\u0AD6\u0AD7\u0AD8\u0AD9\u0ADA\u0ADB\u0ADC\u0ADD\u0ADE\u0ADF\u0AE0\u0AE1\u0AE2\u0AE3\u0AE4\u0AE5\u0AE6\u0AE7\u0AE8\u0AE9\u0AEA\u0AEB\u0AEC\u0AED\u0AEE\u0AEF\u0AF0\u0AF1\u0AF2\u0AF3\u0AF4\u0AF5\u0AF6\u0AF7\u0AF8\u0AF9\u0AFA\u0AFB\u0AFC\u0AFD\u0AFE\u0AFF\u0B00\u0B01\u0B02\u0B03\u0B04\u0B05\u0B06\u0B07\u0B08\u0B09\u0B0A\u0B0B\u0B0C\u0B0D\u0B0E\u0B0F\u0B10\u0B11\u0B12\u0B13\u0B14\u0B15\u0B16\u0B17\u0B18\u0B19\u0B1A\u0B1B\u0B1C\u0B1D\u0B1E\u0B1F\u0B20\u0B21\u0B22\u0B23\u0B24\u0B25\u0B26\u0B27\u0B28\u0B29\u0B2A\u0B2B\u0B2C\u0B2D\u0B2E\u0B2F\u0B30\u0B31\u0B32\u0B33\u0B34\u0B35\u0B36\u0B37\u0B38\u0B39\u0B3A\u0B3B\u0B3C\u0B3D\u0B3E\u0B3F\u0B40\u0B41\u0B42\u0B43\u0B44\u0B45\u0B46\u0B47\u0B48\u0B49\u0B4A\u0B4B\u0B4C\u0B4D\u0B4E\u0B4F\u0B50\u0B51\u0B52\u0B53\u0B54\u0B55\u0B56\u0B57\u0B58\u0B59\u0B5A\u0B5B\u0B5C\u0B5D\u0B5E\u0B5F\u0B60\u0B61\u0B62\u0B63\u0B64\u0B65\u0B66\u0B67\u0B68\u0B69\u0B6A\u0B6B\u0B6C\u0B6D\u0B6E\u0B6F\u0B70\u0B71\u0B72\u0B73\u0B74\u0B75\u0B76\u0B77\u0B78\u0B79\u0B7A\u0B7B\u0B7C\u0B7D\u0B7E\u0B7F\u0B80\u0B81\u0B82\u0B83\u0B84\u0B85\u0B86\u0B87\u0B88\u0B89\u0B8A\u0B8B\u0B8C\u0B8D\u0B8E\u0B8F\u0B90\u0B91\u0B92\u0B93\u0B94\u0B95\u0B96\u0B97\u0B98\u0B99\u0B9A\u0B9B\u0B9C\u0B9D\u0B9E\u0B9F\u0BA0\u0BA1\u0BA2\u0BA3\u0BA4\u0BA5\u0BA6\u0BA7\u0BA8\u0BA9\u0BAA\u0BAB\u0BAC\u0BAD\u0BAE\u0BAF\u0BB0\u0BB1\u0BB2\u0BB3\u0BB4\u0BB5\u0BB6\u0BB7\u0BB8\u0BB9\u0BBA\u0BBB\u0BBC\u0BBD\u0BBE\u0BBF\u0BC0\u0BC1\u0BC2\u0BC3\u0BC4\u0BC5\u0BC6\u0BC7\u0BC8\u0BC9\u0BCA\u0BCB\u0BCD\u0BCE\u0BCF\u0BD0\u0BD1\u0BD2\u0BD3\u0BD4\u0BD5\u0BD6\u0BD7\u0BD8\u0BD9\u0BDA\u0BDB\u0BDC\u0BDD\u0BDE\u0BDF\u0BE0\u0BE1\u0BE2\u0BE3\u0BE4\u0BE5\u0BE6\u0BE7\u0BE8\u0BE9\u0BEA\u0BEB\u0BEC\u0BED\u0BEF\u0BF0\u0BF1\u0BF2\u0BF3\u0BF4\u0BF5\u0BF6\u0BF7\u0BF8\u0BF9\u0BFA\u0BFB\u0BFC\u0BFD\u0BFE\u0BFF\u0C00\u0C01\u0C02\u0C03\u0C04\u0C05\u0C06\u0C07\u0C08\u0C09\u0C0A\u0C0B\u0C0C\u0C0D\u0C0E\u0C0F\u0C10\u0C11\u0C12\u0C13\u0C14\u0C15\u0C16\u0C17\u0C18\u0C19\u0C1A\u0C1B\u0C1C\u0C1D\u0C1E\u0C1F\u0C20\u0C21\u0C22\u0C23\u0C24\u0C25\u0C26\u0C27\u0C28\u0C29\u0C2A\u0C2B\u0C2C\u0C2D\u0C2E\u0C2F\u0C30\u0C31\u0C32\u0C33\u0
```

```

2273 bmp: '\u0900\u0950\u0953\u0963\u0966\u097F\uA8E0\uA8FD'
2274 },
2275 {
2276   name: 'Duployan',
2277   astral: '\uD82F\uD8C0\uDC6A\uDC70\uDC7C\uDC80\uDC88\uDC90\uDC99\uDC9C\uDC9F'
2278 },
2279 {
2280   name: 'Egyptian_Hieroglyphs',
2281   astral: '\uD80C\uD8C0\uDFFF\uD80D\uD800\uDC2E'
2282 },
2283 {
2284   name: 'Elbasan',
2285   astral: '\uD801\uDD00\uDD27'
2286 },
2287 {
2288   name: 'Ethiopic',
2289   bmp: '\u1200\u1248\u124A\u124D\u1250\u1256\u1258\u125A\u125D\u1260\u1288\u128A\u128D\u1290\u12B0\u12B2\u12B5\u12B8\u12BE\u12C0\u12C2\u12C5\u12C8\u12D6'
2290 },
2291 {
2292   name: 'Georgian',
2293   bmp: '\u10A0\u10C5\u10C7\u10CD\u10D0\u10FA\u10FC\u10FF\u2D00\u2D25\u2D27\u2D2D'
2294 },
2295 {
2296   name: 'Glagolitic',
2297   bmp: '\u2C00\u2C2E\u2C30\u2C5E',
2298   astral: '\uD838\uD8C0\uDC06\uDC08\uDC18\uDC1B\uDC21\uDC23\uDC24\uDC26\uDC2A'
2299 },
2300 {
2301   name: 'Gothic',
2302   astral: '\uD800\uDF30\uDF4A'
2303 },
2304 {
2305   name: 'Grantha',
2306   astral: '\uD804\uDF00\uDF03\uDF05\uDF0C\uDF0F\uDF10\uDF13\uDF28\uDF2A\uDF30\uDF32\uDF33\uDF35\uDF39\uDF3C\uDF44\uDF47\uDF48\uDF4B\uDF4D\uDF50\uDF57\uDF5D'
2307 },
2308 {
2309   name: 'Greek',
2310   bmp: '\u0370\u0373\u0375\u0377\u037A\u037D\u037F\u0384\u0386\u0388\u038A\u038C\u038E\u03A1\u03A3\u03E1\u03F0\u03FF\u1D26\u1D2A\u1D5D\u1D61\u1D66\u1D6A\u1D70\u1D7C\u1D7E\u1D80\u1D8C\u1D9E\u1DA1\u1DA5\u1DA7\u1DA9\u1DBF\u1DBF\u1DC0\u1DC1\u1DC2\u1DC3\u1DC4\u1DC5\u1DC6\u1DC7\u1DC8\u1DC9\u1DDA\u1DDC\u1DDF\u1DE0\u1DE1\u1DE2\u1DE3\u1DE4\u1DE5\u1DE6\u1DE7\u1DE8\u1DE9\u1DEA\u1DEB\u1DEC\u1DED\u1DEE\u1DEF\u1DF0\u1DF1\u1DF2\u1DF3\u1DF4\u1DF5\u1DF6\u1DF7\u1DF8\u1DF9\u1DFA\u1DFB\u1DFC\u1DFD\u1DFE\u1DFF\u1E00\u1E01\u1E02\u1E03\u1E04\u1E05\u1E06\u1E07\u1E08\u1E09\u1E1A\u1E1B\u1E1C\u1E1D\u1E1E\u1E1F\u1E20\u1E21\u1E22\u1E23\u1E24\u1E25\u1E26\u1E27\u1E28\u1E29\u1E30\u1E31\u1E32\u1E33\u1E34\u1E35\u1E36\u1E37\u1E38\u1E39\u1E40\u1E41\u1E42\u1E43\u1E44\u1E45\u1E46\u1E47\u1E48\u1E49\u1E50\u1E51\u1E52\u1E53\u1E54\u1E55\u1E56\u1E57\u1E58\u1E59\u1E60\u1E61\u1E62\u1E63\u1E64\u1E65\u1E66\u1E67\u1E68\u1E69\u1E70\u1E71\u1E72\u1E73\u1E74\u1E75\u1E76\u1E77\u1E78\u1E79\u1E80\u1E81\u1E82\u1E83\u1E84\u1E85\u1E86\u1E87\u1E88\u1E89\u1E90\u1E91\u1E92\u1E93\u1E94\u1E95\u1E96\u1E97\u1E98\u1E99\u1EA0\u1EA1\u1EA2\u1EA3\u1EA4\u1EA5\u1EA6\u1EA7\u1EA8\u1EA9\u1EB0\u1EB1\u1EB2\u1EB3\u1EB4\u1EB5\u1EB6\u1EB7\u1EB8\u1EB9\u1EC0\u1EC1\u1EC2\u1EC3\u1EC4\u1EC5\u1EC6\u1EC7\u1EC8\u1EC9\u1ED0\u1ED1\u1ED2\u1ED3\u1ED4\u1ED5\u1ED6\u1ED7\u1ED8\u1ED9\u1EDA\u1EDB\u1EDC\u1EDD\u1EDE\u1EDF\u1EE0\u1EE1\u1EE2\u1EE3\u1EE4\u1EE5\u1EE6\u1EE7\u1EE8\u1EE9\u1EEA\u1EEB\u1EEC\u1EED\u1EEF\u1EF0\u1EF1\u1EF2\u1EF3\u1EF4\u1EF5\u1EF6\u1EF7\u1EF8\u1EF9\u1F00\u1F01\u1F02\u1F03\u1F04\u1F05\u1F06\u1F07\u1F08\u1F09\u1F10\u1F11\u1F12\u1F13\u1F14\u1F15\u1F16\u1F17\u1F18\u1F19\u1F20\u1F21\u1F22\u1F23\u1F24\u1F25\u1F26\u1F27\u1F28\u1F29\u1F30\u1F31\u1F32\u1F33\u1F34\u1F35\u1F36\u1F37\u1F38\u1F39\u1F40\u1F41\u1F42\u1F43\u1F44\u1F45\u1F46\u1F47\u1F48\u1F49\u1F50\u1F51\u1F52\u1F53\u1F54\u1F55\u1F56\u1F57\u1F58\u1F59\u1F60\u1F61\u1F62\u1F63\u1F64\u1F65\u1F66\u1F67\u1F68\u1F69\u1F70\u1F71\u1F72\u1F73\u1F74\u1F75\u1F76\u1F77\u1F78\u1F79\u1F80\u1F81\u1F82\u1F83\u1F84\u1F85\u1F86\u1F87\u1F88\u1F89\u1F90\u1F91\u1F92\u1F93\u1F94\u1F95\u1F96\u1F97\u1F98\u1F99\u1FA0\u1FA1\u1FA2\u1FA3\u1FA4\u1FA5\u1FA6\u1FA7\u1FA8\u1FA9\u1FB0\u1FB1\u1FB2\u1FB3\u1FB4\u1FB5\u1FB6\u1FB7\u1FB8\u1FB9\u1FC0\u1FC1\u1FC2\u1FC3\u1FC4\u1FC5\u1FC6\u1FC7\u1FC8\u1FC9\u1FDA\u1FDB\u1FDC\u1FDD\u1FDE\u1FDF\u1FE0\u1FE1\u1FE2\u1FE3\u1FE4\u1FE5\u1FE6\u1FE7\u1FE8\u1FE9\u1FF0\u1FF1\u1FF2\u1FF3\u1FF4\u1FF5\u1FF6\u1FF7\u1FF8\u1FF9\u1FFA\u1FFB\u1FFC\u1FFD\u1FFE\u1FFF'
2311   astral: '\uD800\uDD40\uDD8E\uDDA0\uDD834\uDE00\uDE45'
2312 },
2313 {
2314   name: 'Gujarati',
2315   bmp: '\u0A81\u0A83\u0A85\u0A8D\u0A8F\u0A91\u0A93\u0AA8\u0AAA\u0AB0\u0AB2\u0AB3\u0AB5\u0AB9\u0ABC\u0AC5\u0AC7\u0AC9\u0ACB\u0ACD\u0AD0\u0AE0\u0AE3\u0AE'
2316 },
2317 {
2318   name: 'Gurmukhi',
2319   bmp: '\u0A01\u0A03\u0A05\u0A0A\u0A0F\u0A10\u0A13\u0A28\u0A2A\u0A30\u0A32\u0A33\u0A35\u0A36\u0A38\u0A39\u0A3C\u0A3E\u0A42\u0A47\u0A48\u0A4B\u0A4D\u0A51\u0'
2320 },
2321 {
2322   name: 'Han',
2323   bmp: '\u2E80\u2E99\u2E9B\u2EF3\u2F00\u2FD5\u3005\u3007\u3021\u3029\u3038\u303B\u3400\u4DB5\u4E00\u9FD5\uF900\uFA6D\uFA70\uFAD9'
2324   astral: '\uD840\uD868\uD86A\uD86C\uD86F\uD872\uDC00\uDFFF\uD869\uDC00\uDED6\uDF00\uDFFF\uD86D\uDC00\uDF34\uDF40\uDFFF\uD86E\uDC00\uDC1D\uDC20\u'
2325 },
2326 {
2327   name: 'Hangul',
2328   bmp: '\u1100\u11FF\u302E\u302F\u3131\u318E\u3200\u321E\u3260\u327E\uA960\uA97C\uAC00\uD7A3\uD7B0\uD7C6\uD7CB\uD7FB\uFFA0\uFFB8\uFFC2\uFFC7\uFFCA\uFFCF'
2329 },
2330 {
2331   name: 'Hanunoo',
2332   bmp: '\u1720\u1734'
2333 },
2334 {
2335   name: 'Hatran',
2336   astral: '\uD802\uDCE0\uDCF2\uDCF4\uDCF5\uDCFB\uDCFF'
2337 },
2338 {
2339   name: 'Hebrew',
2340   bmp: '\u0591\u05C7\u05D0\u05EA\u05F0\u05F4\uFB1D\uFB36\uFB38\uFB3C\uFB3E\uFB40\uFB41\uFB43\uFB44\uFB46\uFB4F'
2341 },
2342 {
2343   name: 'Hiragana',
2344   bmp: '\u3041\u3096\u309D\u309F'
2345   astral: '\uD82C\uDC01\uD83C\uDE00'
2346 },
2347 {
2348   name: 'Imperial_Aramaic',
2349   astral: '\uD802\uDC40\uDC55\uDC57\uDC5F'
2350 },
2351 {
2352   name: 'Inherited',
2353   bmp: '\u0300\u036F\u0485\u0486\u064B\u0655\u0670\u0951\u0952\u1AB0\u1ABE\u1CD0\u1CD2\u1CD4\u1CE0\u1CE2\u1CE8\u1CED\u1CF4\u1CF8\u1CF9\u1DC0\u1DF5\u1DFB\u'
2354   astral: '\uD800\uD8DF\uDEE0\uD834\uDD67\uDD69\uDD7B\uDD82\uDD85\uDD8B\uDDAA\uDDAD\uDB40\uDD00\uDDEF'
2355 },
2356 {
2357   name: 'Inscriptional_Pahlavi',
2358   astral: '\uD802\uDF60\uDF72\uDF78\uDF7F'
2359 },
2360 {
2361   name: 'Inscriptional_Parthian',
2362   astral: '\uD802\uDF40\uDF55\uDF58\uDF5F'
2363 },
2364 {
2365   name: 'Javanese',
2366   bmp: '\uA980\uA9CD\u
```

```

2388 },
2389 {
2390   name: 'Khmer',
2391   bmp: '\u1780\u17DD\u17E0\u17E9\u17F0\u17F9\u19E0\u19FF'
2392 },
2393 {
2394   name: 'Khojki',
2395   astral: '\uD804[\uDE00-\uDE11\uDE13-\uDE3E]'
2396 },
2397 {
2398   name: 'Khudawadi',
2399   astral: '\uD804[\uDEB0-\uDEEA\uDEF0-\uDEF9]'
2400 },
2401 {
2402   name: 'Lao',
2403   bmp: '\u0E81\u0E82\u0E84\u0E87\u0E88\u0E8A\u0E8D\u0E94\u0E97\u0E99\u0E9F\u0EA1\u0EA3\u0EA5\u0EA7\u0EAA\u0EAB\u0EAD\u0EB9\u0EBB\u0EBD\u0EC0\u0EC4\u0EC6\u0E0E'
2404 },
2405 {
2406   name: 'Latin',
2407   bmp: 'A-Za-z\uAA\uBA\uC0-\uD6\uD8-\uF6\uF8-\u02B8\u02E0\u02E4\u1D00\u1D25\u1D2C-\u1D5C\u1D62-\u1D65\u1D6B\u1D77\u1D79\u1DBE\u1E00\u1EFF\u2071\u207F\u2090\u2092\u2093\u2095\u2096\u2098\u2099\u209A\u209C\u209D\u209E\u209F\u2100\u2101\u2102\u2103\u2104\u2105\u2106\u2107\u2108\u2109\u210A\u210B\u210C\u210D\u210E\u210F\u2110\u2111\u2112\u2113\u2114\u2115\u2116\u2117\u2118\u2119\u211A\u211B\u211C\u211D\u211E\u211F\u2120\u2121\u2122\u2123\u2124\u2125\u2126\u2127\u2128\u2129\u212A\u212B\u212C\u212D\u212E\u212F\u2130\u2131\u2132\u2133\u2134\u2135\u2136\u2137\u2138\u2139\u213A\u213B\u213C\u213D\u213E\u213F\u2140\u2141\u2142\u2143\u2144\u2145\u2146\u2147\u2148\u2149\u214A\u214B\u214C\u214D\u214E\u214F\u2150\u2151\u2152\u2153\u2154\u2155\u2156\u2157\u2158\u2159\u215A\u215B\u215C\u215D\u215E\u215F\u2160\u2161\u2162\u2163\u2164\u2165\u2166\u2167\u2168\u2169\u216A\u216B\u216C\u216D\u216E\u216F\u2170\u2171\u2172\u2173\u2174\u2175\u2176\u2177\u2178\u2179\u217A\u217B\u217C\u217D\u217E\u217F\u2180\u2181\u2182\u2183\u2184\u2185\u2186\u2187\u2188\u2189\u218A\u218B\u218C\u218D\u218E\u218F\u2190\u2191\u2192\u2193\u2194\u2195\u2196\u2197\u2198\u2199\u219A\u219B\u219C\u219D\u219E\u219F\u21A0\u21A1\u21A2\u21A3\u21A4\u21A5\u21A6\u21A7\u21A8\u21A9\u21AA\u21AB\u21AC\u21AD\u21AE\u21AF\u21B0\u21B1\u21B2\u21B3\u21B4\u21B5\u21B6\u21B7\u21B8\u21B9\u21BA\u21BB\u21BC\u21BD\u21BE\u21BF\u21C0\u21C1\u21C2\u21C3\u21C4\u21C5\u21C6\u21C7\u21C8\u21C9\u21CA\u21CB\u21CC\u21CD\u21CE\u21CF\u21D0\u21D1\u21D2\u21D3\u21D4\u21D5\u21D6\u21D7\u21D8\u21D9\u21DA\u21DB\u21DC\u21DD\u21DE\u21DF\u21E0\u21E1\u21E2\u21E3\u21E4\u21E5\u21E6\u21E7\u21E8\u21E9\u21EA\u21EB\u21EC\u21ED\u21EE\u21EF\u21F0\u21F1\u21F2\u21F3\u21F4\u21F5\u21F6\u21F7\u21F8\u21F9\u21FA\u21FB\u21FC\u21FD\u21FE\u21FF\u2200\u2201\u2202\u2203\u2204\u2205\u2206\u2207\u2208\u2209\u220A\u220B\u220C\u220D\u220E\u220F\u2210\u2211\u2212\u2213\u2214\u2215\u2216\u2217\u2218\u2219\u221A\u221B\u221C\u221D\u221E\u221F\u2220\u2221\u2222\u2223\u2224\u2225\u2226\u2227\u2228\u2229\u222A\u222B\u222C\u222D\u222E\u222F\u2230\u2231\u2232\u2233\u2234\u2235\u2236\u2237\u2238\u2239\u223A\u223B\u223C\u223D\u223E\u223F\u2240\u2241\u2242\u2243\u2244\u2245\u2246\u2247\u2248\u2249\u224A\u224B\u224C\u224D\u224E\u224F\u2250\u2251\u2252\u2253\u2254\u2255\u2256\u2257\u2258\u2259\u225A\u225B\u225C\u225D\u225E\u225F\u2260\u2261\u2262\u2263\u2264\u2265\u2266\u2267\u2268\u2269\u226A\u226B\u226C\u226D\u226E\u226F\u2270\u2271\u2272\u2273\u2274\u2275\u2276\u2277\u2278\u2279\u227A\u227B\u227C\u227D\u227E\u227F\u2280\u2281\u2282\u2283\u2284\u2285\u2286\u2287\u2288\u2289\u228A\u228B\u228C\u228D\u228E\u228F\u2290\u2291\u2292\u2293\u2294\u2295\u2296\u2297\u2298\u2299\u229A\u229B\u229C\u229D\u229E\u229F\u22A0\u22A1\u22A2\u22A3\u22A4\u22A5\u22A6\u22A7\u22A8\u22A9\u22AA\u22AB\u22AC\u22AD\u22AE\u22AF\u22B0\u22B1\u22B2\u22B3\u22B4\u22B5\u22B6\u22B7\u22B8\u22B9\u22BA\u22BB\u22BC\u22BD\u22BE\u22BF\u22C0\u22C1\u22C2\u22C3\u22C4\u22C5\u22C6\u22C7\u22C8\u22C9\u22CA\u22CB\u22CC\u22CD\u22CE\u22CF\u22D0\u22D1\u22D2\u22D3\u22D4\u22D5\u22D6\u22D7\u22D8\u22D9\u22DA\u22DB\u22DC\u22DD\u22DE\u22DF\u22E0\u22E1\u22E2\u22E3\u22E4\u22E5\u22E6\u22E7\u22E8\u22E9\u22EA\u22EB\u22EC\u22ED\u22EE\u22EF\u22F0\u22F1\u22F2\u22F3\u22F4\u22F5\u22F6\u22F7\u22F8\u22F9\u22FA\u22FB\u22FC\u22FD\u22FE\u22FF\u2280\u2281\u2282\u2283\u2284\u2285\u2286\u2287\u2288\u2289\u228A\u228B\u228C\u228D\u228E\u228F\u2290\u2291\u2292\u2293\u2294\u2295\u2296\u2297\u2298\u2299\u229A\u229B\u229C\u229D\u229E\u229F\u22A0\u22A1\u22A2\u22A3\u22A4\u22A5\u22A6\u22A7\u22A8\u22A9\u22AA\u22AB\u22AC\u22AD\u22AE\u22AF\u22B0\u22B1\u22B2\u22B3\u22B4\u22B5\u22B6\u22B7\u22B8\u22B9\u22BA\u22BB\u22BC\u22BD\u22BE\u22BF\u22C0\u22C1\u22C2\u22C3\u22C4\u22C5\u22C6\u22C7\u22C8\u22C9\u22CA\u22CB\u22CC\u22CD\u22CE\u22CF\u22D0\u22D1\u22D2\u22D3\u22D4\u22D5\u22D6\u22D7\u22D8\u22D9\u22DA\u22DB\u22DC\u22DD\u22DE\u22DF\u22E0\u22E1\u22E2\u22E3\u22E4\u22E5\u22E6\u22E7\u22E8\u22E9\u22EA\u22EB\u22EC\u22ED\u22EE\u22EF\u22F0\u22F1\u22F2\u22F3\u22F4\u22F5\u22F6\u22F7\u22F8\u22F9\u22FA\u22FB\u22FC\u22FD\u22FE\u22FF\u2280\u2281\u2282\u2283\u2284\u2285\u2286\u2287\u2288\u2289\u228A\u228B\u228C\u228D\u228E\u228F\u2290\u2291\u2292\u2293\u2294\u2295\u2296\u2297\u2298\u2299\u229A\u229B\u229C\u229D\u229E\u229F\u22A0\u22A1\u22A2\u22A3\u22A4\u22A5\u22A6\u22A7\u22A8\u22A9\u22AA\u22AB\u22AC\u22AD\u22AE\u22AF\u22B0\u22B1\u22B2\u22B3\u22B4\u22B5\u22B6\u22B7\u22B8\u22B9\u22BA\u22BB\u22BC\u22BD\u22BE\u22BF\u22C0\u22C1\u22C2\u22C3\u22C4\u22C5\u22C6\u22C7\u22C8\u22C9\u22CA\u22CB\u22CC\u22CD\u22CE\u22CF\u22D0\u22D1\u22D2\u22D3\u22D4\u22D5\u22D6\u22D7\u22D8\u22D9\u22DA\u22DB\u22DC\u22DD\u22DE\u22DF\u22E0\u22E1\u2
```

```

2503     name: 'New_Tai_Lue',
2504     bmp: '\u1980-\u19AB\u19B0-\u19C9\u19D0-\u19DA\u19DE\u19DF'
2505 },
2506 {
2507     name: 'Newa',
2508     astral: '\uD805[\uDC00-\uDC59\uDC5B\uDC5D]'
2509 },
2510 {
2511     name: 'Nko',
2512     bmp: '\u07C0-\u07FA'
2513 },
2514 {
2515     name: 'Ogham',
2516     bmp: '\u1680-\u169C'
2517 },
2518 {
2519     name: 'Ol_Chiki',
2520     bmp: '\u1C50-\u1C7F'
2521 },
2522 {
2523     name: 'Old_Hungarian',
2524     astral: '\uD803[\uDC80-\uDCB2\uDCC0-\uDCF2\uDCFA-\uDCFF]'
2525 },
2526 {
2527     name: 'Old_Italic',
2528     astral: '\uD800[\uDF00-\uDF23]'
2529 },
2530 {
2531     name: 'Old_North_Arabian',
2532     astral: '\uD802[\uDE80-\uDE9F]'
2533 },
2534 {
2535     name: 'Old_Permic',
2536     astral: '\uD800[\uDF50-\uDF7A]'
2537 },
2538 {
2539     name: 'Old_Persian',
2540     astral: '\uD800[\uDFA0-\uDFC3\uDFC8-\uDFD5]'
2541 },
2542 {
2543     name: 'Old_South_Arabian',
2544     astral: '\uD802[\uDE60-\uDE7F]'
2545 },
2546 {
2547     name: 'Old_Turkic',
2548     astral: '\uD803[\uDC00-\uDC48]'
2549 },
2550 {
2551     name: 'Oriya',
2552     bmp: '\u0B01-\u0B03\u0B05-\u0B0C\u0B0F\u0B10\u0B13-\u0B28\u0B2A-\u0B30\u0B32\u0B33\u0B35-\u0B39\u0B3C-\u0B44\u0B47\u0B48\u0B4B-\u0B4D\u0B56\u0B57\u0B5C\u0B5D\u0B5F-\u0B60\u0B62\u0B63\u0B65-\u0B6C\u0B6F\u0B70\u0B72\u0B74\u0B76\u0B7C\u0B7D\u0B7F-\u0B80\u0B82\u0B83\u0B85-\u0B8C\u0B8F\u0B90\u0B92\u0B94\u0B96\u0B9C\u0B9D\u0B9F-\u0BA0\u0BA2\u0BA3\u0BA5-\u0BA8\u0BAF\u0BB0\u0BB2\u0BB3\u0BB5-\u0BB9\u0BBB\u0BBF\u0BC0\u0BC2\u0BC3\u0BC5-\u0BCA\u0BCD\u0BCF\u0BD0\u0BD2\u0BD3\u0BD5-\u0BD8\u0BD9\u0BDB\u0BDD\u0BDE\u0BDF\u0BE0\u0BE2\u0BE3\u0BE5-\u0BE8\u0BE9\u0BED\u0BED\u0BEE\u0BEF\u0BF0\u0BF2\u0BF3\u0BF5-\u0BFB\u0BFD\u0C00\u0C02\u0C03\u0C05-\u0C08\u0C09\u0C0D\u0C0E\u0C0F\u0C10\u0C12\u0C13\u0C15-\u0C18\u0C19\u0C1D\u0C1E\u0C1F\u0C20\u0C22\u0C23\u0C25-\u0C28\u0C29\u0C2D\u0C2E\u0C2F\u0C30\u0C32\u0C33\u0C35-\u0C38\u0C39\u0C3D\u0C3E\u0C3F\u0C40\u0C42\u0C43\u0C45-\u0C48\u0C49\u0C4D\u0C4E\u0C4F\u0C50\u0C52\u0C53\u0C55-\u0C58\u0C59\u0C5D\u0C5E\u0C5F\u0C60\u0C62\u0C63\u0C65-\u0C68\u0C69\u0C6D\u0C6E\u0C6F\u0C70\u0C72\u0C73\u0C75-\u0C78\u0C79\u0C7D\u0C7E\u0C7F\u0C80\u0C82\u0C83\u0C85-\u0C88\u0C89\u0C8D\u0C8E\u0C8F\u0C90\u0C92\u0C93\u0C95-\u0C98\u0C99\u0C9D\u0C9E\u0C9F\u0CA0\u0CA2\u0CA3\u0CA5-\u0CA8\u0CA9\u0CAD\u0CAE\u0CAF\u0CB0\u0CB2\u0CB3\u0CB5-\u0CB8\u0CB9\u0CBD\u0CBE\u0CBF\u0CC0\u0CC2\u0CC3\u0CC5-\u0CC8\u0CC9\u0CCD\u0CCE\u0CCF\u0CD0\u0CD2\u0CD3\u0CD5-\u0CD8\u0CD9\u0CDD\u0CDE\u0CDF\u0CE0\u0CE2\u0CE3\u0CE5-\u0CE8\u0CE9\u0CED\u0CEE\u0CEF\u0CF0\u0CF2\u0CF3\u0CF5-\u0CF8\u0CF9\u0CFD\u0CFE\u0CFF\u0D00\u0D02\u0D03\u0D05-\u0D08\u0D09\u0D0D\u0D0E\u0D0F\u0D10\u0D12\u0D13\u0D15-\u0D18\u0D19\u0D1D\u0D1E\u0D1F\u0D20\u0D22\u0D23\u0D25-\u0D28\u0D29\u0D2D\u0D2E\u0D2F\u0D30\u0D32\u0D33\u0D35-\u0D38\u0D39\u0D3D\u0D3E\u0D3F\u0D40\u0D42\u0D43\u0D45-\u0D48\u0D49\u0D4D\u0D4E\u0D4F\u0D50\u0D52\u0D53\u0D55-\u0D58\u0D59\u0D5D\u0D5E\u0D5F\u0D60\u0D62\u0D63\u0D65-\u0D68\u0D69\u0D6D\u0D6E\u0D6F\u0D70\u0D72\u0D73\u0D75-\u0D78\u0D79\u0D7D\u0D7E\u0D7F\u0D80\u0D82\u0D83\u0D85-\u0D88\u0D89\u0D8D\u0D8E\u0D8F\u0D90\u0D92\u0D93\u0D95-\u0D98\u0D99\u0D9D\u0D9E\u0D9F\u0DA0\u0DA2\u0DA3\u0DA5-\u0DA8\u0DA9\u0DAD\u0DAE\u0DAF\u0DB0\u0DB2\u0DB3\u0DB5-\u0DB8\u0DB9\u0DBD\u0DBE\u0DBF\u0DC0\u0DC2\u0DC3\u0DC5-\u0DC8\u0DC9\u0DCD\u0DCE\u0DCF\u0DD0\u0DD2\u0DD3\u0DD5-\u0DD8\u0DD9\u0DDD\u0DDE\u0DDF\u0DE0\u0DE2\u0DE3\u0DE5-\u0DE8\u0DE9\u0DED\u0DEE\u0DEF\u0DF0\u0DF2\u0DF3\u0DF5-\u0DF8\u0DF9\u0DFD\u0DFE\u0DFF\u0E00\u0E02\u0E03\u0E05-\u0E08\u0E09\u0E0D\u0E0E\u0E0F\u0E10\u0E12\u0E13\u0E15-\u0E18\u0E19\u0E1D\u0E1E\u0E1F\u0E20\u0E22\u0E23\u0E25-\u0E28\u0E29\u0E2D\u0E2E\u0E2F\u0E30\u0E32\u0E33\u0E35-\u0E38\u0E39\u0E3D\u0E3E\u0E3F\u0E40\u0E42\u0E43\u0E45-\u0E48\u0E49\u0E4D\u0E4E\u0E4F\u0E50\u0E52\u0E53\u0E55-\u0E58\u0E59\u0E5D\u0E5E\u0E5F\u0E60\u0E62\u0E63\u0E65-\u0E68\u0E69\u0E6D\u0E6E\u0E6F\u0E70\u0E72\u0E73\u0E75-\u0E78\u0E79\u0E7D\u0E7E\u0E7F\u0E80\u0E82\u0E83\u0E85-\u0E88\u0E89\u0E8D\u0E8E\u0E8F\u0E90\u0E92\u0E93\u0E95-\u0E98\u0E99\u0E9D\u0E9E\u0E9F\u0EA0\u0EA2\u0EA3\u0EA5-\u0EA8\u0EA9\u0EAD\u0EAE\u0EAF\u0EB0\u0EB2\u0EB3\u0EB5-\u0EB8\u0EB9\u0EBD\u0EBE\u0EBF\u0EC0\u0EC2\u0EC3\u0EC5-\u0EC8\u0EC9\u0ECD\u0ECE\u0ECF\u0ED0\u0ED2\u0ED3\u0ED5-\u0ED8\u0ED9\u0EDD\u0EDE\u0EDF\u0EE0\u0EE2\u0EE3\u0EE5-\u0EE8\u0EE9\u0EED\u0EEE\u0EEF\u0EF0\u0EF2\u0EF3\u0EF5-\u0EF8\u0EF9\u0EFD\u0EFE\u0EFF\u0F00\u0F02\u0F03\u0F05-\u0F08\u0F09\u0F0D\u0F0E\u0F0F\u0F10\u0F12\u0F13\u0F15-\u0F18\u0F19\u0F1D\u0F1E\u0F1F\u0F20\u0F22\u0F23\u0F25-\u0F28\u0F29\u0F2D\u0F2E\u0F2F\u0F30\u0F32\u0F33\u0F35-\u0F38\u0F39\u0F3D\u0F3E\u0F3F\u0F40\u0F42\u0F43\u0F45-\u0F48\u0F49\u0F4D\u0F4E\u0F4F\u0F50\u0F52\u0F53\u0F55-\u0F58\u0F59\u0F5D\u0F5E\u0F5F\u0F60\u0F62\u0F63\u0F65-\u0F68\u0F69\u0F6D\u0F6E\u0F6F\u0F70\u0F72\u0F73\u0F75-\u0F78\u0F79\u0F7D\u0F7E\u0F7F\u0F80\u0F82\u0F83\u0F85-\u0F88\u0F89\u0F8D\u0F8E\u0F8F\u0F90\u0F92\u0F93\u0F95-\u0F98\u0F99\u0F9D\u0F9E\u0F9F\u0FA0\u0FA2\u0FA3\u0FA5-\u0FA8\u0FA9\u0FAD\u0FAE\u0FAF\u0FB0\u0FB2\u0FB3\u0FB5-\u0FB8\u0FB9\u0FBD\u0FBE\u0FBF\u0FC0\u0FC2\u0FC3\u0FC5-\u0FC8\u0FC9\u0FCD\u0FCE\u0FCF\u0FD0\u0FD2\u0FD3\u0FD5-\u0FD8\u0FD9\u0FDD\u0FDE\u0FDF\u0FE0\u0FE2\u0FE3\u0FE5-\u0FE8\u0FE9\u0FED\u0FEE\u0FEF\u0FF0\u0FF2\u0FF3\u0FF5-\u0FF8\u0FF9\u0FFD\u0FFE\u0FFF'
2553 },
2554 {
2555     name: 'Osage',
2556     astral: '\uD801[\uDCB0-\uDCD3\uDCD8-\uDCFB]'
2557 },
2558 {
2559     name: 'Osmanya',
2560     astral: '\uD801[\uDC80-\uDC9D\uDCA0-\uDCA9]'
2561 },
2562 {
2563     name: 'Pahawh_Hmong',
2564     astral: '\uD81A[\uDF00-\uDF45\uDF50-\uDF59\uDF5B-\uDF61\uDF63-\uDF77\uDF7D-\uDF8F]'
2565 },
2566 {
2567     name: 'Palmyrene',
2568     astral: '\uD802[\uDC60-\uDC7F]'
2569 },
2570 {
2571     name: 'Pau_Cin_Hau',
2572     astral: '\uD806[\uDEC0-\uDEF8]'
2573 },
2574 {
2575     name: 'Phags_Pa',
2576     bmp: '\uA840-\uA877'
2577 },
2578 {
2579     name: 'Phoenician',
2580     astral: '\uD802[\uDD00-\uDD1B\uDD1F]'
2581 },
2582 {
2583     name: 'Psalter_Pahlavi',
2584     astral: '\uD802[\uDF80-\uDF99\uDF9C\uDFA9-\uDFAF]'
2585 },
2586 {
2587     name: 'Rejang',
2588     bmp: '\uA930-\uA953\uA95F'
2589 },
2590 {
2591     name: 'Runic',
2592     bmp: '\u16A0-\u16EA\u16EE-\u16F8'
2593 },
2594 {
2595     name: 'Samaritan',
2596     bmp: '\u0800-\u082D\u0830-\u083E'
2597 },
2598 {
2599     name: 'Saurashtra',
2600     bmp: '\uA880-\uA8C5\uA8CE-\uA8D9'
2601 },
2602 {
2603     name: 'Sharada',
2604     astral: '\uD804[\uDD80-\uDDCD\uDDDD\uDDDF]'
2605 },
2606 {
2607     name: 'Shavian',
2608     astral: '\uD801[\uDC50-\uDC7F]'
2609 },
2610 {
2611     name: 'Siddham',
2612     astral: '\uD805[\uDD80-\uDDDB\uDDDB\uDDDD]'
2613 },
2614 {
2615     name: 'SignWriting',
2616     astral: '\uD836[\uDC00-\uDE8B\uDE9B-\uDE9F\uDEA1-\uDEAF]'
2617 },

```

```

2618     {
2619       name: 'Sinhala',
2620       bmp: '\u0D82\u0D83\u0D85-\u0D96\u0D9A-\u0DB1\u0DB3-\u0DBB\u0DBD\u0DC0-\u0DC6\u0DCA\u0DCF-\u0DD4\u0DD6\u0DD8-\u0DDF\u0DE6-\u0DEF\u0DF2-\u0DF4',
2621       astral: '\uD804[\uDDE1-\uDDF4]'
2622     },
2623     {
2624       name: 'Sora_Sompeng',
2625       astral: '\uD804[\uDCD0-\uDCE8\uDCF0-\uDCF9]'
2626     },
2627     {
2628       name: 'Sundanese',
2629       bmp: '\u1B80-\u1BBF\u1CC0-\u1CC7'
2630     },
2631     {
2632       name: 'Syloiti_Nagri',
2633       bmp: '\uA800-\uA82B'
2634     },
2635     {
2636       name: 'Syriac',
2637       bmp: '\u0700-\u070D\u070F-\u074A\u074D-\u074F'
2638     },
2639     {
2640       name: 'Tagalog',
2641       bmp: '\u1700-\u170C\u170E-\u1714'
2642     },
2643     {
2644       name: 'Tagbanwa',
2645       bmp: '\u1760-\u176C\u176E-\u1770\u1772\u1773'
2646     },
2647     {
2648       name: 'Tai_Le',
2649       bmp: '\u1950-\u196D\u1970-\u1974'
2650     },
2651     {
2652       name: 'Tai_Tham',
2653       bmp: '\u1A20-\u1A5E\u1A60-\u1A7C\u1A7F-\u1A89\u1A90-\u1A99\u1AA0-\u1AAD'
2654     },
2655     {
2656       name: 'Tai_Viet',
2657       bmp: '\uAA80-\uAAC2\uAADB-\uAADF'
2658     },
2659     {
2660       name: 'Takri',
2661       astral: '\uD805[\uDE80-\uDEB7\uDEC0-\uDEC9]'
2662     },
2663     {
2664       name: 'Tamil',
2665       bmp: '\u0B82\u0B83\u0B85-\u0B8A\u0B8E-\u0B90\u0B92-\u0B95\u0B99\u0B9A\u0B9C\u0B9E\u0B9F\u0BA3\u0BA4\u0BA8-\u0BAA\u0BAE-\u0BB9\u0BBE-\u0BC2\u0BC6-\u0BC8\u0BCA-\u0BCD\u0BCF'
2666     },
2667     {
2668       name: 'Tangut',
2669       astral: '\uD81B\uDFE0[\uD81C-\uDB20][\uDC00-\uDFFF]\uD821[\uDC00-\uDFEC]\uD822[\uDC00-\uDEF2]'
2670     },
2671     {
2672       name: 'Telugu',
2673       bmp: '\u0C00-\u0C03\u0C05-\u0C0C\u0C0E-\u0C10\u0C12-\u0C28\u0C2A-\u0C39\u0C3D-\u0C44\u0C46-\u0C48\u0C4A-\u0C4D\u0C55\u0C56\u0C58-\u0C5A\u0C60-\u0C63\u0C66-\u0C6F'
2674     },
2675     {
2676       name: 'Thaana',
2677       bmp: '\u0780-\u07B1'
2678     },
2679     {
2680       name: 'Thai',
2681       bmp: '\u0E01-\u0E3A\u0E40-\u0E5B'
2682     },
2683     {
2684       name: 'Tibetan',
2685       bmp: '\u0F00-\u0F47\u0F49-\u0F6C\u0F71-\u0F97\u0F99-\u0FBC\u0FBE-\u0FCC\u0FCE-\u0FD4\u0FD9\u0FDA'
2686     },
2687     {
2688       name: 'Tifinagh',
2689       bmp: '\u2D30-\u2D67\u2D6F\u2D70\u2D7F'
2690     },
2691     {
2692       name: 'Tirhuta',
2693       astral: '\uD805[\uDC80-\uDCC7\uDCD0-\uDCD9]'
2694     },
2695     {
2696       name: 'Ugaritic',
2697       astral: '\uD800[\uDF80-\uDF9D\uDF9F]'
2698     },
2699     {
2700       name: 'Vai',
2701       bmp: '\uA500-\uA62B'
2702     },
2703     {
2704       name: 'Warang_Citi',
2705       astral: '\uD806[\uDCA0-\uDCF2\uDCFF]'
2706     },
2707     {
2708       name: 'Yi',
2709       bmp: '\uA000-\uA48C\uA490-\uA4C6'
2710     }
2711   ]]);
2712
2713 };
2714
2715 },{}].8:[function(require,module,exports){
2716   var XRegExp = require('./xregexp');
2717
2718   require('./addons/build')(XRegExp);
2719   require('./addons/matchrecursive')(XRegExp);
2720   require('./addons/unicode-base')(XRegExp);
2721   require('./addons/unicode-blocks')(XRegExp);
2722   require('./addons/unicode-categories')(XRegExp);
2723   require('./addons/unicode-properties')(XRegExp);
2724   require('./addons/unicode-scripts')(XRegExp);
2725
2726   module.exports = XRegExp;
2727
2728   },{}],"/addons/build".1,"/addons/matchrecursive".2,"/addons/unicode-base".3,"/addons/unicode-blocks".4,"/addons/unicode-categories".5,"/addons/unicode-properties".6,"/addons/unicode-scripts".7,"/addons/unicode-props".8,"/addons/unicode-props".9,"/addons/unicode-props".10,"/addons/unicode-props".11,"/addons/unicode-props".12,"/addons/unicode-props".13,"/addons/unicode-props".14,"/addons/unicode-props".15,"/addons/unicode-props".16,"/addons/unicode-props".17,"/addons/unicode-props".18,"/addons/unicode-props".19,"/addons/unicode-props".20,"/addons/unicode-props".21,"/addons/unicode-props".22,"/addons/unicode-props".23,"/addons/unicode-props".24,"/addons/unicode-props".25,"/addons/unicode-props".26,"/addons/unicode-props".27,"/addons/unicode-props".28,"/addons/unicode-props".29,"/addons/unicode-props".30,"/addons/unicode-props".31,"/addons/unicode-props".32,"/addons/unicode-props".33,"/addons/unicode-props".34,"/addons/unicode-props".35,"/addons/unicode-props".36,"/addons/unicode-props".37,"/addons/unicode-props".38,"/addons/unicode-props".39,"/addons/unicode-props".40,"/addons/unicode-props".41,"/addons/unicode-props".42,"/addons/unicode-props".43,"/addons/unicode-props".44,"/addons/unicode-props".45,"/addons/unicode-props".46,"/addons/unicode-props".47,"/addons/unicode-props".48,"/addons/unicode-props".49,"/addons/unicode-props".50,"/addons/unicode-props".51,"/addons/unicode-props".52,"/addons/unicode-props".53,"/addons/unicode-props".54,"/addons/unicode-props".55,"/addons/unicode-props".56,"/addons/unicode-props".57,"/addons/unicode-props".58,"/addons/unicode-props".59,"/addons/unicode-props".60,"/addons/unicode-props".61,"/addons/unicode-props".62,"/addons/unicode-props".63,"/addons/unicode-props".64,"/addons/unicode-props".65,"/addons/unicode-props".66,"/addons/unicode-props".67,"/addons/unicode-props".68,"/addons/unicode-props".69,"/addons/unicode-props".70,"/addons/unicode-props".71,"/addons/unicode-props".72,"/addons/unicode-props".73,"/addons/unicode-props".74,"/addons/unicode-props".75,"/addons/unicode-props".76,"/addons/unicode-props".77,"/addons/unicode-props".78,"/addons/unicode-props".79,"/addons/unicode-props".80,"/addons/unicode-props".81,"/addons/unicode-props".82,"/addons/unicode-props".83,"/addons/unicode-props".84,"/addons/unicode-props".85,"/addons/unicode-props".86,"/addons/unicode-props".87,"/addons/unicode-props".88,"/addons/unicode-props".89,"/addons/unicode-props".90,"/addons/unicode-props".91,"/addons/unicode-props".92,"/addons/unicode-props".93,"/addons/unicode-props".94,"/addons/unicode-props".95,"/addons/unicode-props".96,"/addons/unicode-props".97,"/addons/unicode-props".98,"/addons/unicode-props".99,"/addons/unicode-props".100,"/addons/unicode-props".101,"/addons/unicode-props".102,"/addons/unicode-props".103,"/addons/unicode-props".104,"/addons/unicode-props".105,"/addons/unicode-props".106,"/addons/unicode-props".107,"/addons/unicode-props".108,"/addons/unicode-props".109,"/addons/unicode-props".110,"/addons/unicode-props".111,"/addons/unicode-props".112,"/addons/unicode-props".113,"/addons/unicode-props".114,"/addons/unicode-props".115,"/addons/unicode-props".116,"/addons/unicode-props".117,"/addons/unicode-props".118,"/addons/unicode-props".119,"/addons/unicode-props".120,"/addons/unicode-props".121,"/addons/unicode-props".122,"/addons/unicode-props".123,"/addons/unicode-props".124,"/addons/unicode-props".125,"/addons/unicode-props".126,"/addons/unicode-props".127,"/addons/unicode-props".128,"/addons/unicode-props".129,"/addons/unicode-props".130,"/addons/unicode-props".131,"/addons/unicode-props".132,"/addons/unicode-props".133,"/addons/unicode-props".134,"/addons/unicode-props".135,"/addons/unicode-props".136,"/addons/unicode-props".137,"/addons/unicode-props".138,"/addons/unicode-props".139,"/addons/unicode-props".140,"/addons/unicode-props".141,"/addons/unicode-props".142,"/addons/unicode-props".143,"/addons/unicode-props".144,"/addons/unicode-props".145,"/addons/unicode-props".146,"/addons/unicode-props".147,"/addons/unicode-props".148,"/addons/unicode-props".149,"/addons/unicode-props".150,"/addons/unicode-props".151,"/addons/unicode-props".152,"/addons/unicode-props".153,"/addons/unicode-props".154,"/addons/unicode-props".155,"/addons/unicode-props".156,"/addons/unicode-props".157,"/addons/unicode-props".158,"/addons/unicode-props".159,"/addons/unicode-props".160,"/addons/unicode-props".161,"/addons/unicode-props".162,"/addons/unicode-props".163,"/addons/unicode-props".164,"/addons/unicode-props".165,"/addons/unicode-props".166,"/addons/unicode-props".167,"/addons/unicode-props".168,"/addons/unicode-props".169,"/addons/unicode-props".170,"/addons/unicode-props".171,"/addons/unicode-props".172,"/addons/unicode-props".173,"/addons/unicode-props".174,"/addons/unicode-props".175,"/addons/unicode-props".176,"/addons/unicode-props".177,"/addons/unicode-props".178,"/addons/unicode-props".179,"/addons/unicode-props".180,"/addons/unicode-props".181,"/addons/unicode-props".182,"/addons/unicode-props".183,"/addons/unicode-props".184,"/addons/unicode-props".185,"/addons/unicode-props".186,"/addons/unicode-props".187,"/addons/unicode-props".188,"/addons/unicode-props".189,"/addons/unicode-props".190,"/addons/unicode-props".191,"/addons/unicode-props".192,"/addons/unicode-props".193,"/addons/unicode-props".194,"/addons/unicode-props".195,"/addons/unicode-props".196,"/addons/unicode-props".197,"/addons/unicode-props".198,"/addons/unicode-props".199,"/addons/unicode-props".200,"/addons/unicode-props".201,"/addons/unicode-props".202,"/addons/unicode-props".203,"/addons/unicode-props".204,"/addons/unicode-props".205,"/addons/unicode
```



```

2733 */
2734
2735 'use strict';
2736
2737 /**
2738 * XRegExp provides augmented, extensible regular expressions. You get additional regex syntax and
2739 * flags, beyond what browsers support natively. XRegExp is also a regex utility belt with tools to
2740 * make your client-side grepping simpler and more powerful, while freeing you from related
2741 * cross-browser inconsistencies.
2742 */
2743
2744 // =====
2745 // Private stuff
2746 // =====
2747
2748 // Property name used for extended regex instance data
2749 var REGEX_DATA = 'xregexp';
2750 // Optional features that can be installed and uninstalled
2751 var features = {
2752   astral: false,
2753   natives: false
2754 };
2755 // Native methods to use and restore ('native' is an ES3 reserved keyword)
2756 var nativ = {
2757   exec: RegExp.prototype.exec,
2758   test: RegExp.prototype.test,
2759   match: String.prototype.match,
2760   replace: String.prototype.replace,
2761   split: String.prototype.split
2762 };
2763 // Storage for fixed/extended native methods
2764 var fixed = {};
2765 // Storage for regexes cached by `XRegExp.cache`
2766 var regexCache = {};
2767 // Storage for pattern details cached by the `XRegExp` constructor
2768 var patternCache = {};
2769 // Storage for regex syntax tokens added internally or by `XRegExp.addToken`
2770 var tokens = [];
2771 // Token scopes
2772 var defaultScope = 'default';
2773 var classScope = 'class';
2774 // Regexes that match native regex syntax, including octals
2775 var nativeTokens = {
2776   // Any native multicharacter token in default scope, or any single character
2777   'default': [/(?[0-3][0-7]{0,2}||[4-7][0-7]?|1-9\d*x|dA Fa f|{2}|u(?:dA Fa f|{2})|c[A Za z]|S|(?[:|=|]|<|=|)|[?*+]|d+(?:d*)?|?|S|/,
2778   // Any native multicharacter token in character class scope, or any single character
2779   'class': [/(?[0-3][0-7]{0,2}[4-7][0-7]?|x|dA Fa f|{2}|u(?:dA Fa f|{2})|c[A Za z]|S|(?[:|=|]|<|=|)|[?*+]|d+(?:d*)?|?|S|/,
2780 ];
2781 // Any backreference or dollar-prefixed character in replacement strings
2782 var replacementToken = /\$(?:(\w$)+)|(\d\d?|S)/g;
2783 // Check for correct `exec` handling of nonparticipating capturing groups
2784 var correctExecNpcg = nativ.exec.call(/()??/, "")[1] === undefined;
2785 // Check for ES6 `flags` prop support
2786 var hasFlagsProp = /x/ flags === undefined;
2787 // Shortcut to `Object.prototype.toString`
2788 var toString = {}.toString;
2789
2790 function hasNativeFlag(flag) {
2791   // Can't check based on the presence of properties/getters since browsers might support such
2792   // properties even when they don't support the corresponding flag in regex construction (tested
2793   // in Chrome 48, where `unicode` in /x/ is true but trying to construct a regex with flag `u`
2794   // throws an error)
2795   var isSupported = true;
2796   try {
2797     // Can't use regex literals for testing even in a `try` because regex literals with
2798     // unsupported flags cause a compilation error in IE
2799     new RegExp("", flag);
2800   } catch (exception) {
2801     isSupported = false;
2802   }
2803   return isSupported;
2804 }
2805 // Check for ES6 `u` flag support
2806 var hasNativeU = hasNativeFlag('u');
2807 // Check for ES6 `y` flag support
2808 var hasNativeY = hasNativeFlag('y');
2809 // Tracker for known flags, including addon flags
2810 var registeredFlags = {
2811   g: true,
2812   i: true,
2813   m: true,
2814   u: hasNativeU,
2815   y: hasNativeY
2816 };
2817
2818 /**
2819 * Attaches extended data and `XRegExp.prototype` properties to a regex object.
2820 *
2821 * @private
2822 * @param {RegExp} regex Regex to augment.
2823 * @param {Array} captureNames Array with capture names, or `null`.
2824 * @param {String} xSource XRegExp pattern used to generate `regex`, or `null` if N/A.
2825 * @param {String} xFlags XRegExp flags used to generate `regex`, or `null` if N/A.
2826 * @param {Boolean} [isInternalOnly=false] Whether the regex will be used only for internal
2827 * operations, and never exposed to users. For internal-only regexes, we can improve perf by
2828 * skipping some operations like attaching `XRegExp.prototype` properties.
2829 * @returns {RegExp} Augmented regex.
2830 */
2831 function augment(regex, captureNames, xSource, xFlags, isInternalOnly) {
2832   var p;
2833
2834   regex[REGEX_DATA] = {
2835     captureNames: captureNames
2836   };
2837
2838   if (isInternalOnly) {
2839     return regex;
2840   }
2841
2842   // Can't auto-inherit these since the XRegExp constructor returns a nonprimitive value
2843   if (regex.__proto__) {
2844     regex.__proto__ = XRegExp.prototype;
2845   } else {
2846     for (p in XRegExp.prototype) {
2847       // An `XRegExp.prototype.hasOwnProperty(p)` check wouldn't be worth it here, since this

```

```

2848     // is performance sensitive, and enumerable `Object.prototype` or `RegExp.prototype`
2849     // extensions exist on `regex.prototype` anyway
2850     regex.p = XRegExp.prototype.p];
2851 }
2852 }
2853
2854 regex|REGEX_DATA| source = xSource;
2855 // Emulate the ES6 `flags` prop by ensuring flags are in alphabetical order
2856 regex|REGEX_DATA| flags = xFlags ? xFlags.split("").sort().join("") : xFlags;
2857
2858 return regex;
2859 }
2860
2861 /**
2862  * Removes any duplicate characters from the provided string.
2863  *
2864  * @private
2865  * @param {String} str String to remove duplicate characters from.
2866  * @returns {String} String with any duplicate characters removed.
2867  */
2868 function clipDuplicates(str) {
2869     return nativ.replace.call(str, /([{sS})?(?=[{sS}]*\1)/g, "");
2870 }
2871
2872 /**
2873  * Copies a regex object while preserving extended data and augmenting with `XRegExp.prototype`
2874  * properties. The copy has a fresh `lastIndex` property (set to zero). Allows adding and removing
2875  * flags g and y while copying the regex.
2876  *
2877  * @private
2878  * @param {RegExp} regex Regex to copy.
2879  * @param {Object} [options] Options object with optional properties:
2880  *   - `addG` {Boolean} Add flag g while copying the regex.
2881  *   - `addY` {Boolean} Add flag y while copying the regex.
2882  *   - `removeG` {Boolean} Remove flag g while copying the regex.
2883  *   - `removeY` {Boolean} Remove flag y while copying the regex.
2884  *   - `isInternalOnly` {Boolean} Whether the copied regex will be used only for internal
2885  *     operations, and never exposed to users. For internal-only regexes, we can improve perf by
2886  *     skipping some operations like attaching `XRegExp.prototype` properties.
2887  *   - `source` {String} Overrides `<regex>.source`, for special cases.
2888  * @returns {RegExp} Copy of the provided regex, possibly with modified flags.
2889  */
2890 function copyRegex(regex, options) {
2891     if (!XRegExp.isRegExp(regex)) {
2892         throw new TypeError("Type RegExp expected");
2893     }
2894
2895     var xData = regex|REGEX_DATA| || {};
2896     var flags = getNativeFlags|regex|;
2897     var flagsToAdd = "";
2898     var flagsToRemove = "";
2899     var xregexpSource = null;
2900     var xregexpFlags = null;
2901
2902     options = options || {};
2903
2904     if (options.removeG) { flagsToRemove += 'g'; }
2905     if (options.removeY) { flagsToRemove += 'y'; }
2906     if (flagsToRemove) {
2907         flags = nativ.replace.call(flags, new RegExp('[' + flagsToRemove + ']+', 'g'), "");
2908     }
2909
2910     if (options.addG) { flagsToAdd += 'g'; }
2911     if (options.addY) { flagsToAdd += 'y'; }
2912     if (flagsToAdd) {
2913         flags = clipDuplicates(flags + flagsToAdd);
2914     }
2915
2916     if (options.isInternalOnly) {
2917         if (xData.source !== undefined) {
2918             xregexpSource = xData.source;
2919         }
2920         // null or undefined; don't want to add to `flags` if the previous value was null, since
2921         // that indicates we're not tracking original precompilation flags
2922         if (xData.flags !== null) {
2923             // Flags are only added for non-internal regexes by `XRegExp.globalize`. Flags are never
2924             // removed for non-internal regexes, so don't need to handle it
2925             xregexpFlags = flagsToAdd + clipDuplicates(xData.flags + flagsToAdd) : xData.flags;
2926         }
2927     }
2928
2929     // Augment with `XRegExp.prototype` properties, but use the native `RegExp` constructor to avoid
2930     // searching for special tokens. That would be wrong for regexes constructed by `RegExp`, and
2931     // unnecessary for regexes constructed by `XRegExp` because the regex has already undergone the
2932     // translation to native regex syntax
2933     regex = augment(
2934         new RegExp(options.source || regex.source, flags),
2935         hasNamedCapture|regex| ? xData.captureNames.slice(0) : null,
2936         xregexpSource,
2937         xregexpFlags,
2938         options.isInternalOnly
2939     );
2940
2941     return regex;
2942 }
2943
2944 /**
2945  * Converts hexadecimal to decimal.
2946  *
2947  * @private
2948  * @param {String} hex
2949  * @returns {Number}
2950  */
2951 function dec(hex) {
2952     return parseInt(hex, 16);
2953 }
2954
2955 /**
2956  * Returns a pattern that can be used in a native RegExp in place of an ignorable token such as an
2957  * inline comment or whitespace with flag x. This is used directly as a token handler function
2958  * passed to `XRegExp.addToken`.
2959  *
2960  * @private
2961  * @param {String} match Match arg of `XRegExp.addToken` handler
2962  * @param {String} scope Scope arg of `XRegExp.addToken` handler

```

```

2963 * @param {String} flags Flags arg of `XRegExp.addToken` handler
2964 * @returns {String} Either '' or '(?:)', depending on which is needed in the context of the match.
2965 */
2966 function getContextualTokenSeparator(match, scope, flags) {
2967     if (
2968         // No need to separate tokens if at the beginning or end of a group
2969         match.input.charAt(match.index - 1) === '(' ||
2970         match.input.charAt(match.index + match[0].length) === ')' ||
2971         // Avoid separating tokens when the following token is a quantifier
2972         isPatternNext(match.input, match.index + match[0].length, flags, '[?+*]{\\d+(?:,\\d*)?}')
2973     ) {
2974         return '';
2975     }
2976     // Keep tokens separated. This avoids e.g. inadvertently changing `\\1 1` or `\\1(?:#)1` to `\\1\\1`.
2977     // This also ensures all tokens remain as discrete atoms, e.g. it avoids converting the syntax
2978     // error `(?:` into `(?:`.
2979     return '(?:)';
2980 }
2981
2982 /**
2983 * Returns native `RegExp` flags used by a regex object.
2984 *
2985 * @private
2986 * @param {RegExp} regex Regex to check.
2987 * @returns {String} Native flags in use.
2988 */
2989 function getNativeFlags(regex) {
2990     return hasFlagsProp ?
2991         regex.flags :
2992         // Explicitly using `RegExp.prototype.toString` (rather than e.g. `String` or concatenation
2993         // with an empty string) allows this to continue working predictably when
2994         // `XRegExp.prototype.toString` is overridden
2995         nativ.exec.call(/([a-z]*)$/i, RegExp.prototype.toString.call(regex))[1];
2996 }
2997
2998 /**
2999 * Determines whether a regex has extended instance data used to track capture names.
3000 *
3001 * @private
3002 * @param {RegExp} regex Regex to check.
3003 * @returns {Boolean} Whether the regex uses named capture.
3004 */
3005 function hasNamedCapture(regex) {
3006     return !!((regex.REGEX_DATA) && regex.REGEX_DATA.captureNames);
3007 }
3008
3009 /**
3010 * Converts decimal to hexadecimal.
3011 *
3012 * @private
3013 * @param {Number|String} dec
3014 * @returns {String}
3015 */
3016 function hex(dec) {
3017     return parseInt(dec, 10).toString(16);
3018 }
3019
3020 /**
3021 * Returns the first index at which a given value can be found in an array.
3022 *
3023 * @private
3024 * @param {Array} array Array to search.
3025 * @param {*} value Value to locate in the array.
3026 * @returns {Number} Zero-based index at which the item is found, or -1.
3027 */
3028 function indexOf(array, value) {
3029     var len = array.length;
3030     var i;
3031     for (i = 0; i < len; ++i) {
3032         if (array[i] === value) {
3033             return i;
3034         }
3035     }
3036     return -1;
3037 }
3038
3039 /**
3040 * Checks whether the next nonignorable token after the specified position matches the
3041 * `needlePattern`
3042 *
3043 * @private
3044 *
3045 * @param {String} pattern Pattern to search within.
3046 * @param {Number} pos Index in `pattern` to search at.
3047 * @param {String} flags Flags used by the pattern.
3048 * @param {String} needlePattern Pattern to match the next token against.
3049 * @returns {Boolean} Whether the next nonignorable token matches `needlePattern`
3050 */
3051 function isPatternNext(pattern, pos, flags, needlePattern) {
3052     var inlineCommentPattern = '\\(\\?#[^)]*\\)';
3053     var lineCommentPattern = '#(?:\\n)*';
3054     var patternsToIgnore = flags.indexOf('x') > -1 ?
3055         // Ignore any leading whitespace, line comments, and inline comments
3056         ['\\s', lineCommentPattern, inlineCommentPattern] :
3057         // Ignore any leading inline comments
3058         [inlineCommentPattern];
3059     return nativ.test.call(
3060         new RegExp('^(?:' + patternsToIgnore.join('|') + ')*(?:' + needlePattern + ')'),
3061         pattern.slice(pos)
3062     );
3063 }
3064
3065 /**
3066 * Determines whether a value is of the specified type, by resolving its internal [[Class]].
3067 *
3068 * @private
3069 * @param {*} value Object to check.
3070 * @param {String} type Type to check for, in TitleCase.
3071 * @returns {Boolean} Whether the object matches the type.
3072 */
3073 function isType(value, type) {
3074     return toString.call(value) === '[object ' + type + ']';
3075 }
3076
3077

```

```

3078 /**
3079  * Adds leading zeros if shorter than four characters. Used for fixed-length hexadecimal values.
3080  *
3081  * @private
3082  * @param {String} str
3083  * @returns {String}
3084  */
3085 function pad4(str) {
3086   while (str.length < 4) {
3087     str = '0' + str;
3088   }
3089   return str;
3090 }
3091
3092 /**
3093  * Checks for flag-related errors, and strips/applies flags in a leading mode modifier. Offloads
3094  * the flag preparation logic from the `XRegExp` constructor.
3095  *
3096  * @private
3097  * @param {String} pattern Regex pattern, possibly with a leading mode modifier.
3098  * @param {String} flags Any combination of flags.
3099  * @returns {Object} Object with properties `pattern` and `flags`.
3100  */
3101 function prepareFlags(pattern, flags) {
3102   var i;
3103
3104   // Recent browsers throw on duplicate flags, so copy this behavior for nonnative flags
3105   if (clipDuplicates(flags) !== flags) {
3106     throw new SyntaxError('Invalid duplicate regex flag ' + flags);
3107   }
3108
3109   // Strip and apply a leading mode modifier with any combination of flags except g or y
3110   pattern = nativ.replace.call(pattern, /^(\?([ws]+))\)/, function($0, $1) {
3111     if (nativ.test.call(/(gy)/, $1)) {
3112       throw new SyntaxError('Cannot use flag g or y in mode modifier ' + $0);
3113     }
3114     // Allow duplicate flags within the mode modifier
3115     flags = clipDuplicates(flags + $1);
3116     return '';
3117   });
3118
3119   // Throw on unknown native or nonnative flags
3120   for (i = 0; i < flags.length; ++i) {
3121     if (!registeredFlags[flags.charAt(i)]) {
3122       throw new SyntaxError('Unknown regex flag ' + flags.charAt(i));
3123     }
3124   }
3125
3126   return {
3127     pattern: pattern,
3128     flags: flags
3129   };
3130 }
3131
3132 /**
3133  * Prepares an options object from the given value.
3134  *
3135  * @private
3136  * @param {String|Object} value Value to convert to an options object.
3137  * @returns {Object} Options object.
3138  */
3139 function prepareOptions(value) {
3140   var options = {};
3141
3142   if (isType(value, 'String')) {
3143     XRegExp.forEach(value, /^[^\s,]+/, function(match) {
3144       options[match] = true;
3145     });
3146   }
3147   return options;
3148 }
3149
3150 return value;
3151 }
3152
3153 /**
3154  * Registers a flag so it doesn't throw an 'unknown flag' error.
3155  *
3156  * @private
3157  * @param {String} flag Single-character flag to register.
3158  */
3159 function registerFlag(flag) {
3160   if (!/^[ws,]+$/i.test(flag)) {
3161     throw new Error('Flag must be a single character A-Za-z0-9_$');
3162   }
3163
3164   registeredFlags[flag] = true;
3165 }
3166
3167 /**
3168  * Runs built-in and custom regex syntax tokens in reverse insertion order at the specified
3169  * position, until a match is found.
3170  *
3171  * @private
3172  * @param {String} pattern Original pattern from which an XRegExp object is being built.
3173  * @param {String} flags Flags being used to construct the regex.
3174  * @param {Number} pos Position to search for tokens within `pattern`.
3175  * @param {Number} scope Regex scope to apply: 'default' or 'class'.
3176  * @param {Object} context Context object to use for token handler functions.
3177  * @returns {Object} Object with properties `matchLength`, `output`, and `reparse`; or `null`.
3178  */
3179 function runTokens(pattern, flags, pos, scope, context) {
3180   var i = tokens.length;
3181   var leadChar = pattern.charAt(pos);
3182   var result = null;
3183   var match;
3184   var t;
3185
3186   // Run in reverse insertion order
3187   while (i--) {
3188     t = tokens[i];
3189     if (
3190       (t.leadChar && t.leadChar !== leadChar) ||
3191       (t.scope !== scope && t.scope !== 'all') ||
3192       (t.flag && flags.indexOf(t.flag) === -1)

```

```

3193     } {
3194         continue;
3195     }
3196
3197     match = XRegExp.exec(pattern, t.regex, pos, 'sticky');
3198     if (match) {
3199         result = {
3200             matchLength: match[0].length,
3201             output: t.handler.call(context, match, scope, flags),
3202             reparse: t.reparse
3203         };
3204         // Finished with token tests
3205         break;
3206     }
3207 }
3208
3209 return result;
3210 }
3211
3212 /**
3213  * Enables or disables implicit astral mode opt-in. When enabled, flag A is automatically added to
3214  * all new regexes created by XRegExp. This causes an error to be thrown when creating regexes if
3215  * the Unicode Base addon is not available, since flag A is registered by that addon.
3216  *
3217  * @private
3218  * @param {Boolean} on `true` to enable; `false` to disable.
3219  */
3220 function setAstral(on) {
3221     features astral = on;
3222 }
3223
3224 /**
3225  * Enables or disables native method overrides.
3226  *
3227  * @private
3228  * @param {Boolean} on `true` to enable; `false` to disable.
3229  */
3230 function setNatives(on) {
3231     RegExp.prototype.exec = (on ? fixed : native).exec;
3232     RegExp.prototype.test = (on ? fixed : native).test;
3233     String.prototype.match = (on ? fixed : native).match;
3234     String.prototype.replace = (on ? fixed : native).replace;
3235     String.prototype.split = (on ? fixed : native).split;
3236
3237     features natives = on;
3238 }
3239
3240 /**
3241  * Returns the object, or throws an error if it is `null` or `undefined`. This is used to follow
3242  * the ES5 abstract operation `ToObject`.
3243  *
3244  * @private
3245  * @param {*} value Object to check and return.
3246  * @returns {*} The provided object.
3247  */
3248 function toObject(value) {
3249     // null or undefined
3250     if (value == null) {
3251         throw new TypeError('Cannot convert null or undefined to object');
3252     }
3253
3254     return value;
3255 }
3256
3257 // =====
3258 // Constructor
3259 // =====
3260
3261 /**
3262  * Creates an extended regular expression object for matching text with a pattern. Differs from a
3263  * native regular expression in that additional syntax and flags are supported. The returned object
3264  * is in fact a native `RegExp` and works with all native methods.
3265  *
3266  * @class XRegExp
3267  * @constructor
3268  * @param {String|RegExp} pattern Regex pattern string, or an existing regex object to copy.
3269  * @param {String} [flags] Any combination of flags.
3270  * Native flags:
3271  * - `g` - global
3272  * - `i` - ignore case
3273  * - `m` - multiline anchors
3274  * - `u` - unicode (ES6)
3275  * - `y` - sticky (Firefox 3+, ES6)
3276  * Additional XRegExp flags:
3277  * - `n` - explicit capture
3278  * - `s` - dot matches all (aka singleline)
3279  * - `x` - free-spacing and line comments (aka extended)
3280  * - `A` - astral (requires the Unicode Base addon)
3281  * Flags cannot be provided when constructing one `RegExp` from another.
3282  * @returns {RegExp} Extended regular expression object.
3283  * @example
3284  *
3285  * // With named capture and flag x
3286  * XRegExp('(?!<year> [0-9]{4}) -? # year \n\
3287  *         (?!<month> [0-9]{2}) -? # month \n\
3288  *         (?!<day> [0-9]{2}) # day ', 'x');
3289  *
3290  * // Providing a regex object copies it. Native regexes are recompiled using native (not XRegExp)
3291  * // syntax. Copies maintain extended data, are augmented with `XRegExp.prototype` properties, and
3292  * // have fresh `lastIndex` properties (set to zero).
3293  * XRegExp(/regex/);
3294  */
3295 function XRegExp(pattern, flags) {
3296     if (XRegExp.isRegExp(pattern)) {
3297         if (flags !== undefined) {
3298             throw new TypeError('Cannot supply flags when copying a RegExp');
3299         }
3300         return copyRegex(pattern);
3301     }
3302
3303     // Copy the argument behavior of `RegExp`
3304     pattern = pattern === undefined ? '' : String(pattern);
3305     flags = flags === undefined ? '' : String(flags);
3306
3307     if (XRegExp.isInstalled('astral') && flags.indexOf('A') === -1) {

```

```

3308 // This causes an error to be thrown if the Unicode Base addon is not available
3309 flags += 'A';
3310 }
3311
3312 if (!patternCache.pattern) {
3313     patternCache.pattern = {};
3314 }
3315
3316 if (!patternCache.pattern[flags]) {
3317     var context = {
3318         hasNamedCapture: false,
3319         captureNames: []
3320     };
3321     var scope = defaultScope;
3322     var output = "";
3323     var pos = 0;
3324     var result;
3325
3326     // Check for flag-related errors, and strip/apply flags in a leading mode modifier
3327     var applied = prepareFlags(pattern, flags);
3328     var appliedPattern = applied.pattern;
3329     var appliedFlags = applied.flags;
3330
3331     // Use XRegExp's tokens to translate the pattern to a native regex pattern.
3332     // `appliedPattern.length` may change on each iteration if tokens use `reparse`
3333     while (pos < appliedPattern.length) {
3334         do {
3335             // Check for custom tokens at the current position
3336             result = runTokens(appliedPattern, appliedFlags, pos, scope, context);
3337             // If the matched token used the `reparse` option, splice its output into the
3338             // pattern before running tokens again at the same position
3339             if (result && result.reparse) {
3340                 appliedPattern = appliedPattern.slice(0, pos) +
3341                     result.output +
3342                     appliedPattern.slice(pos + result.matchLength);
3343             }
3344         } while (result && result.reparse);
3345
3346         if (result) {
3347             output += result.output;
3348             pos += (result.matchLength || 1);
3349         } else {
3350             // Get the native token at the current position
3351             var token = XRegExp.exec(appliedPattern, nativeTokens[scope], pos, 'sticky')[0];
3352             output += token;
3353             pos += token.length;
3354             if (token === '[' && scope === defaultScope) {
3355                 scope = classScope;
3356             } else if (token === '[' && scope === classScope) {
3357                 scope = defaultScope;
3358             }
3359         }
3360     }
3361
3362     patternCache.pattern[flags] = {
3363         // Use basic cleanup to collapse repeated empty groups like `(?:)(?:)` to `(?:)`. Empty
3364         // groups are sometimes inserted during regex transpilation in order to keep tokens
3365         // separated. However, more than one empty group in a row is never needed.
3366         pattern: native.replace.call(output, /(?:\(\?\:\/)+/g, '(?:)'),
3367         // Strip all but native flags
3368         flags: native.replace.call(appliedFlags, /(?:^gimuy)+/g, ""),
3369         // `context.captureNames` has an item for each capturing group, even if unnamed
3370         captures: context.hasNamedCapture ? context.captureNames : null
3371     };
3372 }
3373
3374 var generated = patternCache.pattern[flags];
3375 return augment(
3376     new RegExp(generated.pattern, generated.flags),
3377     generated.captures,
3378     pattern,
3379     flags
3380 );
3381 }
3382
3383 // Add `RegExp.prototype` to the prototype chain
3384 XRegExp.prototype = new RegExp();
3385
3386 // =====
3387 // Public properties
3388 // =====
3389
3390 /**
3391  * The XRegExp version number as a string containing three dot-separated parts. For example,
3392  * '2.0.0-beta-3'.
3393  */
3394 * @static
3395 * @memberOf XRegExp
3396 * @type String
3397 */
3398 XRegExp.version = '3.2.0';
3399
3400 // =====
3401 // Public methods
3402 // =====
3403
3404 // Intentionally undocumented; used in tests and addons
3405 XRegExp._clipDuplicates = clipDuplicates;
3406 XRegExp._hasNativeFlag = hasNativeFlag;
3407 XRegExp._dec = dec;
3408 XRegExp._hex = hex;
3409 XRegExp._pad4 = pad4;
3410
3411 /**
3412  * Extends XRegExp syntax and allows custom flags. This is used internally and can be used to
3413  * create XRegExp addons. If more than one token can match the same string, the last added wins.
3414  */
3415 * @memberOf XRegExp
3416 * @param {RegExp} regex Regex object that matches the new token.
3417 * @param {Function} handler Function that returns a new pattern string (using native regex syntax)
3418 *   to replace the matched token within all future XRegExp regexes. Has access to persistent
3419 *   properties of the regex being built, through `this`. Invoked with three arguments:
3420 *   - The match array, with named backreference properties.
3421 *   - The regex scope where the match was found: 'default' or 'class'.
3422 *   - The flags used by the regex, including any flags in a leading mode modifier.

```

```

3423 * The handler function becomes part of the XRegExp construction process, so be careful not to
3424 * construct XRegExp objects within the function or you will trigger infinite recursion.
3425 * @param {Object} [options] Options object with optional properties:
3426 * - `scope` {String} Scope where the token applies: 'default', 'class', or 'all'.
3427 * - `flag` {String} Single-character flag that triggers the token. This also registers the
3428 *   flag, which prevents XRegExp from throwing an 'unknown flag' error when the flag is used.
3429 * - `optionalFlags` {String} Any custom flags checked for within the token `handler` that are
3430 *   not required to trigger the token. This registers the flags, to prevent XRegExp from
3431 *   throwing an 'unknown flag' error when any of the flags are used.
3432 * - `reparse` {Boolean} Whether the `handler` function's output should not be treated as
3433 *   final, and instead be reparseable by other tokens (including the current token). Allows
3434 *   token chaining or deferring.
3435 * - `leadChar` {String} Single character that occurs at the beginning of any successful match
3436 *   of the token (not always applicable). This doesn't change the behavior of the token unless
3437 *   you provide an erroneous value. However, providing it can increase the token's performance
3438 *   since the token can be skipped at any positions where this character doesn't appear.
3439 * @example
3440 *
3441 * // Basic usage: Add \a for the ALERT control code
3442 * XRegExp.addToken(
3443 *   /\a/,
3444 *   function() {return '\x07';},
3445 *   {scope: 'all'}
3446 * );
3447 * XRegExp('\a[\a-\n]+').test('\x07\n\x07'); // -> true
3448 *
3449 * // Add the U (ungreedy) flag from PCRE and RE2, which reverses greedy and lazy quantifiers.
3450 * // Since `scope` is not specified, it uses 'default' (i.e., transformations apply outside of
3451 * // character classes only)
3452 * XRegExp.addToken(
3453 *   /(?:+){(d+(?:;d*)?)}(?:)?/,
3454 *   function(match) {return match[1] + (match[2] ? " : '?'");},
3455 *   {flag: 'U'}
3456 * );
3457 * XRegExp('a+', 'U').exec('aaa')[0]; // -> 'a'
3458 * XRegExp('a+?', 'U').exec('aaa')[0]; // -> 'aaa'
3459 */
3460 XRegExp.addToken = function(regex, handler, options) {
3461   options = options || {};
3462   var optionalFlags = options.optionalFlags;
3463   var i;
3464
3465   if (options.flag) {
3466     registerFlag(options.flag);
3467   }
3468
3469   if (optionalFlags) {
3470     optionalFlags = nativeSplitCall(optionalFlags, '');
3471     for (i = 0; i < optionalFlags.length; ++i) {
3472       registerFlag(optionalFlags[i]);
3473     }
3474   }
3475
3476   // Add to the private list of syntax tokens
3477   tokens.push({
3478     regex: copyRegex(regex), {
3479       addG: true,
3480       addY: hasNativeY,
3481       isInternalOnly: true
3482     },
3483     handler: handler,
3484     scope: options.scope || defaultScope,
3485     flag: options.flag,
3486     reparse: options.reparse,
3487     leadChar: options.leadChar
3488   });
3489
3490   // Reset the pattern cache used by the `XRegExp` constructor, since the same pattern and flags
3491   // might now produce different results
3492   XRegExp.cache.flush('patterns');
3493 };
3494
3495 /**
3496 * Caches and returns the result of calling `XRegExp(pattern, flags)`. On any subsequent call with
3497 * the same pattern and flag combination, the cached copy of the regex is returned.
3498 *
3499 * @memberOf XRegExp
3500 * @param {String} pattern Regex pattern string.
3501 * @param {String} [flags] Any combination of XRegExp flags.
3502 * @returns {RegExp} Cached XRegExp object.
3503 * @example
3504 *
3505 * while (match = XRegExp.cache('.', 'gs').exec(str)) {
3506 *   // The regex is compiled once only
3507 * }
3508 */
3509 XRegExp.cache = function(pattern, flags) {
3510   if (!regexCache[pattern]) {
3511     regexCache[pattern] = {};
3512   }
3513   return regexCache[pattern][flags] || (
3514     regexCache[pattern][flags] = XRegExp(pattern, flags)
3515   );
3516 };
3517
3518 // Intentionally undocumented; used in tests
3519 XRegExp.cache.flush = function(cacheName) {
3520   if (cacheName === 'patterns') {
3521     // Flush the pattern cache used by the `XRegExp` constructor
3522     patternCache = {};
3523   } else {
3524     // Flush the regex cache populated by `XRegExp.cache`
3525     regexCache = {};
3526   }
3527 };
3528
3529 /**
3530 * Escapes any regular expression metacharacters, for use when matching literal strings. The result
3531 * can safely be used at any point within a regex that uses any flags.
3532 *
3533 * @memberOf XRegExp
3534 * @param {String} str String to escape.
3535 * @returns {String} String with regex metacharacters escaped.
3536 * @example
3537 *

```

```

3538 * XRegExp.escape('Escaped? <.>');
3539 * // -> 'Escaped?[<.>]
3540 */
3541 XRegExp.escape = function(str) {
3542     return nativ.replace.call(toObject(str), /[-\[\]{}()*+?.,\\^$|#\s]/g, "\\$&");
3543 };
3544
3545 /**
3546 * Executes a regex search in a specified string. Returns a match array or `null`. If the provided
3547 * regex uses named capture, named backreference properties are included on the match array.
3548 * Optional `pos` and `sticky` arguments specify the search start position, and whether the match
3549 * must start at the specified position only. The `lastIndex` property of the provided regex is not
3550 * used, but is updated for compatibility. Also fixes browser bugs compared to the native
3551 * `RegExp.prototype.exec` and can be used reliably cross-browser.
3552 *
3553 * @memberOf XRegExp
3554 * @param {String} str String to search.
3555 * @param {RegExp} regex Regex to search with.
3556 * @param {Number} [pos=0] Zero-based index at which to start the search.
3557 * @param {Boolean|String} [sticky=false] Whether the match must start at the specified position
3558 * only. The string `sticky` is accepted as an alternative to `true`.
3559 * @returns {Array} Match array with named backreference properties, or `null`.
3560 * @example
3561 *
3562 * // Basic use, with named backreference
3563 * var match = XRegExp.exec('U+2620', XRegExp('U\\+(?<hex>[0-9A-F]{4})'));
3564 * match.hex; // -> '2620'
3565 *
3566 * // With pos and sticky, in a loop
3567 * var pos = 2, result = [], match;
3568 * while (match = XRegExp.exec('<1><2><3><4>5<6>', /<(d)>/, pos, 'sticky')) {
3569 *     result.push(match[1]);
3570 *     pos = match.index + match[0].length;
3571 * }
3572 * // result -> ['2', '3', '4']
3573 */
3574 XRegExp.exec = function(str, regex, pos, sticky) {
3575     var cacheKey = 'g';
3576     var addY = false;
3577     var fakeY = false;
3578     var match;
3579     var r2;
3580
3581     addY = hasNativeY && !(sticky || (regex.sticky && sticky !== false));
3582     if (addY) {
3583         cacheKey += 'y';
3584     } else if (sticky) {
3585         // Simulate sticky matching by appending an empty capture to the original regex. The
3586         // resulting regex will succeed no matter what at the current index (set with `lastIndex`),
3587         // and will not search the rest of the subject string. We'll know that the original regex
3588         // has failed if that last capture is `""` rather than `undefined` (i.e., if that last
3589         // capture participated in the match).
3590         fakeY = true;
3591         cacheKey += 'FakeY';
3592     }
3593
3594     regex[REGEX_DATA] = regex[REGEX_DATA] || {};
3595
3596     // Shares cached copies with `XRegExp.match` / `replace`
3597     r2 = regex[REGEX_DATA][cacheKey] || {
3598         regex: REGEX_DATA[cacheKey] = copyRegex(regex, {
3599             addG: true,
3600             addY: addY,
3601             source: fakeY ? regex.source + '()' : undefined,
3602             removeY: sticky === false,
3603             isInternalOnly: true
3604         })
3605     };
3606
3607     pos = pos || 0;
3608     r2.lastIndex = pos;
3609
3610     // Fixed `exec` required for `lastIndex` fix, named backreferences, etc.
3611     match = fixed.exec.call(r2, str);
3612
3613     // Get rid of the capture added by the pseudo-sticky matcher if needed. An empty string means
3614     // the original regexp failed (see above).
3615     if (fakeY && match && match.pop() === "") {
3616         match = null;
3617     }
3618
3619     if (regex.global) {
3620         regex.lastIndex = match ? r2.lastIndex : 0;
3621     }
3622
3623     return match;
3624 };
3625
3626 /**
3627 * Executes a provided function once per regex match. Searches always start at the beginning of the
3628 * string and continue until the end, regardless of the state of the regex's `global` property and
3629 * initial `lastIndex`.
3630 *
3631 * @memberOf XRegExp
3632 * @param {String} str String to search.
3633 * @param {RegExp} regex Regex to search with.
3634 * @param {Function} callback Function to execute for each match. Invoked with four arguments:
3635 * - The match array, with named backreference properties.
3636 * - The zero-based match index.
3637 * - The string being traversed.
3638 * - The regex object being used to traverse the string.
3639 * @example
3640 *
3641 * // Extracts every other digit from a string
3642 * var evens = [];
3643 * XRegExp.forEach('1a2345', /d/, function(match, i) {
3644 *     if (i % 2) evens.push(+match[0]);
3645 * });
3646 * // evens -> [2, 4]
3647 */
3648 XRegExp.forEach = function(str, regex, callback) {
3649     var pos = 0;
3650     var i = -1;
3651     var match;
3652

```



```

3653 while ((match = XRegExp.exec(str, regex, pos))) {
3654     // Because `regex` is provided to `callback`, the function could use the deprecated/
3655     // nonstandard `RegExp.prototype.compile` to mutate the regex. However, since `XRegExp.exec`
3656     // doesn't use `lastIndex` to set the search position, this can't lead to an infinite loop,
3657     // at least. Actually, because of the way `XRegExp.exec` caches globalized versions of
3658     // regexes, mutating the regex will not have any effect on the iteration or matched strings,
3659     // which is a nice side effect that brings extra safety.
3660     callback(match, ++i, str, regex);
3661 }
3662 pos = match.index + (match[0].length || 1);
3663 }
3664 };
3665
3666 /**
3667  * Copies a regex object and adds flag `g`. The copy maintains extended data, is augmented with
3668  * `XRegExp.prototype` properties, and has a fresh `lastIndex` property (set to zero). Native
3669  * regexes are not recompiled using XRegExp syntax.
3670  *
3671  * @memberOf XRegExp
3672  * @param {RegExp} regex Regex to globalize.
3673  * @returns {RegExp} Copy of the provided regex with flag `g` added.
3674  * @example
3675  *
3676  * var globalCopy = XRegExp.globalize(regex);
3677  * globalCopy.global; // -> true
3678  */
3679 XRegExp.globalize = function(regex) {
3680     return copyRegex(regex, {addG: true});
3681 };
3682
3683 /**
3684  * Installs optional features according to the specified options. Can be undone using
3685  * `XRegExp.uninstall`.
3686  *
3687  * @memberOf XRegExp
3688  * @param {Object|String} options Options object or string.
3689  * @example
3690  *
3691  * // With an options object
3692  * XRegExp.install({
3693     * // Enables support for astral code points in Unicode addons (implicitly sets flag A)
3694     * astral: true,
3695     *
3696     * // DEPRECATED: Overrides native regex methods with fixed/extended versions
3697     * natives: true
3698     * });
3699  *
3700  * // With an options string
3701  * XRegExp.install('astral natives');
3702  */
3703 XRegExp.install = function(options) {
3704     options = prepareOptions(options);
3705
3706     if (features.astral && options.astral) {
3707         setAstral(true);
3708     }
3709
3710     if (features.natives && options.natives) {
3711         setNatives(true);
3712     }
3713 };
3714
3715 /**
3716  * Checks whether an individual optional feature is installed.
3717  *
3718  * @memberOf XRegExp
3719  * @param {String} feature Name of the feature to check. One of:
3720  *   - `astral`
3721  *   - `natives`
3722  * @returns {Boolean} Whether the feature is installed.
3723  * @example
3724  *
3725  * XRegExp.isInstalled('astral');
3726  */
3727 XRegExp.isInstalled = function(feature) {
3728     return !!features[feature];
3729 };
3730
3731 /**
3732  * Returns `true` if an object is a regex; `false` if it isn't. This works correctly for regexes
3733  * created in another frame, when `instanceof` and `constructor` checks would fail.
3734  *
3735  * @memberOf XRegExp
3736  * @param {*} value Object to check.
3737  * @returns {Boolean} Whether the object is a `RegExp` object.
3738  * @example
3739  *
3740  * XRegExp.isRegExp('string'); // -> false
3741  * XRegExp.isRegExp(regex/i); // -> true
3742  * XRegExp.isRegExp(RegExp('^', 'm')); // -> true
3743  * XRegExp.isRegExp(XRegExp('?.')); // -> true
3744  */
3745 XRegExp.isRegExp = function(value) {
3746     return toString.call(value) === '[object RegExp]';
3747     //return isType(value, 'RegExp');
3748 };
3749
3750 /**
3751  * Returns the first matched string, or in global mode, an array containing all matched strings.
3752  * This is essentially a more convenient re-implementation of `String.prototype.match` that gives
3753  * the result types you actually want (string instead of `exec`-style array in match-first mode,
3754  * and an empty array instead of `null` when no matches are found in match-all mode). It also lets
3755  * you override flag g and ignore `lastIndex`, and fixes browser bugs.
3756  *
3757  * @memberOf XRegExp
3758  * @param {String} str String to search.
3759  * @param {RegExp} regex Regex to search with.
3760  * @param {String} [scope='one'] Use 'one' to return the first match as a string. Use 'all' to
3761  *   return an array of all matched strings. If not explicitly specified and `regex` uses flag g,
3762  *   `scope` is 'all'.
3763  * @returns {String|Array} In match-first mode: First match as a string, or `null`. In match-all
3764  *   mode: Array of all matched strings, or an empty array.
3765  * @example
3766  *
3767  * // Match first

```

```

3768 * XRegExp.match('abc', /\w/); // -> 'a'
3769 * XRegExp.match('abc', /\w/g, 'one'); // -> 'a'
3770 * XRegExp.match('abc', /\w/g, 'one'); // -> null
3771 *
3772 * // Match all
3773 * XRegExp.match('abc', /\w/g); // -> ['a', 'b', 'c']
3774 * XRegExp.match('abc', /\w/, 'all'); // -> ['a', 'b', 'c']
3775 * XRegExp.match('abc', /\w/, 'all'); // -> []
3776 */
3777 XRegExp.match = function(str, regex, scope) {
3778   var global = (regex.global && scope !== 'one') || scope === 'all';
3779   var cacheKey = (global ? 'g' : '') + (regex.sticky ? 'y' : ''); || 'noGY';
3780   var result;
3781   var r2;
3782
3783   regex[REGEX_DATA] = regex[REGEX_DATA] || {};
3784
3785   // Shares cached copies with `XRegExp.exec`/`replace`
3786   r2 = regex[REGEX_DATA][cacheKey] || (
3787     regex[REGEX_DATA][cacheKey] = copyRegex(regex, {
3788       addG: !global,
3789       removeG: scope === 'one',
3790       isInternalOnly: true
3791     })
3792   );
3793
3794   result = nativeMatch.call(toObject(str), r2);
3795
3796   if (regex.global) {
3797     regex.lastIndex = (
3798       scope === 'one' && result ?
3799         // Can't use `r2.lastIndex` since `r2` is nonglobal in this case
3800         (result.index + result[0].length) : 0
3801     );
3802   }
3803
3804   return global ? (result || []) : (result && result[0]);
3805 };
3806
3807 /**
3808  * Retrieves the matches from searching a string using a chain of regexes that successively search
3809  * within previous matches. The provided `chain` array can contain regexes and or objects with
3810  * `regex` and `backref` properties. When a backreference is specified, the named or numbered
3811  * backreference is passed forward to the next regex or returned.
3812  *
3813  * @memberOf XRegExp
3814  * @param {String} str String to search.
3815  * @param {Array} chain Regexes that each search for matches within preceding results.
3816  * @returns {Array} Matches by the last regex in the chain, or an empty array.
3817  * @example
3818  *
3819  * // Basic usage; matches numbers within <b> tags
3820  * XRegExp.matchChain('1 <b>2</b> 3 <b>4 a 56</b>', [
3821  *   XRegExp('(\\d|is)<b>.*?</b>'),
3822  *   /\\d+/
3823  * ]);
3824  * // -> ['2', '4', '56']
3825  *
3826  * // Passing forward and returning specific backreferences
3827  * html = '<a href="http://xregexp.com/api/">XRegExp</a>\\
3828  * <a href="http://www.google.com/">Google</a>';
3829  * XRegExp.matchChain(html, [
3830  *   {regex: /<a href="(\\^\\+)">/i, backref: 1},
3831  *   {regex: XRegExp('(\\d|https?:\\/\\?<domain>[\\^?#]+)', backref: 'domain')}
3832  * ]);
3833  * // -> ['xregexp.com', 'www.google.com']
3834  */
3835 XRegExp.matchChain = function(str, chain) {
3836   return (function recurseChain(values, level) {
3837     var item = chain[level].regex ? chain[level] : {regex: chain[level]};
3838     var matches = [];
3839
3840     function addMatch(match) {
3841       if (item.backref) {
3842         // Safari 4.0.5 (but not 5.0.5+) inappropriately uses sparse arrays to hold the
3843         // `undefined`'s for backreferences to nonparticipating capturing groups. In such
3844         // cases, a `hasOwnProperty` or `in` check on its own would inappropriately throw
3845         // the exception, so also check if the backreference is a number that is within the
3846         // bounds of the array.
3847         if (!(match.hasOwnProperty(item.backref) || +item.backref < match.length)) {
3848           throw new ReferenceError("Backreference to undefined group: " + item.backref);
3849         }
3850
3851         matches.push(match[item.backref] || "");
3852       } else {
3853         matches.push(match[0]);
3854       }
3855     }
3856
3857     for (var i = 0; i < values.length; ++i) {
3858       XRegExp.forEach(values[i], item.regex, addMatch);
3859     }
3860
3861     return ((level === chain.length - 1) || !matches.length) ?
3862       matches :
3863       recurseChain(matches, level + 1);
3864   })(str, 0);
3865 };
3866
3867 /**
3868  * Returns a new string with one or all matches of a pattern replaced. The pattern can be a string
3869  * or regex, and the replacement can be a string or a function to be called for each match. To
3870  * perform a global search and replace, use the optional `scope` argument or include flag g if using
3871  * a regex. Replacement strings can use `${n}` for named and numbered backreferences. Replacement
3872  * functions can use named backreferences via `arguments[0].name`. Also fixes browser bugs compared
3873  * to the native `String.prototype.replace` and can be used reliably cross-browser.
3874  *
3875  * @memberOf XRegExp
3876  * @param {String} str String to search.
3877  * @param {RegExp|String} search Search pattern to be replaced.
3878  * @param {String|Function} replacement Replacement string or a function invoked to create it.
3879  * Replacement strings can include special replacement syntax:
3880  * - $$ - Inserts a literal $ character.
3881  * - $&, $0 - Inserts the matched substring.
3882  * - `${}` - Inserts the string that precedes the matched substring (left context).

```

```

3883 * - '$' - Inserts the string that follows the matched substring (right context).
3884 * - '$n, $nn' - Where n/nn are digits referencing an existent capturing group, inserts
3885 *   backreference n/nn.
3886 * - '${n}' - Where n is a name or any number of digits that reference an existent capturing
3887 *   group, inserts backreference n.
3888 * Replacement functions are invoked with three or more arguments:
3889 * - The matched substring (corresponds to $& above). Named backreferences are accessible as
3890 *   properties of this first argument.
3891 * - 0..n arguments, one for each backreference (corresponding to $1, $2, etc. above).
3892 * - The zero-based index of the match within the total search string.
3893 * - The total string being searched.
3894 * @param {String} [scope='one'] Use 'one' to replace the first match only, or 'all'. If not
3895 *   explicitly specified and using a regex with flag g, 'scope' is 'all'.
3896 * @returns {String} New string with one or all matches replaced.
3897 * @example
3898 *
3899 * // Regex search, using named backreferences in replacement string
3900 * var name = XRegExp('(?!<first>\\w+) (?!<last>\\w+)');
3901 * XRegExp.replace('John Smith', name, '${last}, ${first}');
3902 * // -> 'Smith, John'
3903 *
3904 * // Regex search, using named backreferences in replacement function
3905 * XRegExp.replace('John Smith', name, function(match) {
3906 *   return match.last + ', ' + match.first;
3907 * });
3908 * // -> 'Smith, John'
3909 *
3910 * // String search, with replace-all
3911 * XRegExp.replace('RegExp builds RegExps', 'RegExp', 'XRegExp', 'all');
3912 * // -> 'XRegExp builds XRegExps'
3913 */
3914 XRegExp.replace = function(str, search, replacement, scope) {
3915   var isRegex = XRegExp.isRegExp(search);
3916   var global = (search.global && scope !== 'one') || scope === 'all';
3917   var cacheKey = ((global ? 'g' : '') + (search.sticky ? 'y' : '')) || 'noGY';
3918   var s2 = search;
3919   var result;
3920
3921   if (isRegex) {
3922     search.REGEX_DATA = search.REGEX_DATA || {};
3923
3924     // Shares cached copies with `XRegExp.exec`'s match`. Since a copy is used, `search`'s
3925     // `lastIndex` isn't updated *during* replacement iterations
3926     s2 = search.REGEX_DATA[cacheKey] || (
3927       search.REGEX_DATA[cacheKey] = copyRegex(search, {
3928         addG: !global,
3929         removeG: scope === 'one',
3930         isInternalOnly: true
3931       })
3932     );
3933   } else if (global) {
3934     s2 = new RegExp(XRegExp.escape(String(search)), 'g');
3935   }
3936
3937   // Fixed `replace` required for named backreferences, etc.
3938   result = fixedReplace.call(toObject(str), s2, replacement);
3939
3940   if (isRegex && search.global) {
3941     // Fixes IE, Safari bug (last tested IE 9, Safari 5.1)
3942     search.lastIndex = 0;
3943   }
3944
3945   return result;
3946 };
3947
3948 /**
3949 * Performs batch processing of string replacements. Used like `XRegExp.replace`, but accepts an
3950 * array of replacement details. Later replacements operate on the output of earlier replacements.
3951 * Replacement details are accepted as an array with a regex or string to search for, the
3952 * replacement string or function, and an optional scope of 'one' or 'all'. Uses the XRegExp
3953 * replacement text syntax, which supports named backreference properties via `${name}`.
3954 *
3955 * @memberOf XRegExp
3956 * @param {String} str String to search.
3957 * @param {Array} replacements Array of replacement detail arrays.
3958 * @returns {String} New string with all replacements.
3959 * @example
3960 *
3961 * str = XRegExp.replaceEach(str, [
3962 *   [XRegExp('(?!<name>a)'), 'z${name}'],
3963 *   [/b/gi, 'y'],
3964 *   [/c/g, 'x', 'one'], // scope 'one' overrides /g
3965 *   [/d/, 'w', 'all'], // scope 'all' overrides lack of /g
3966 *   ['e', 'v', 'all'], // scope 'all' allows replace-all for strings
3967 *   [/f/g, function($0) {
3968 *     return $0.toUpperCase();
3969 *   }]
3970 * ]);
3971 */
3972 XRegExp.replaceEach = function(str, replacements) {
3973   var i;
3974   var r;
3975
3976   for (i = 0; i < replacements.length; ++i) {
3977     r = replacements[i];
3978     str = XRegExp.replace(str, r[0], r[1], r[2]);
3979   }
3980
3981   return str;
3982 };
3983
3984 /**
3985 * Splits a string into an array of strings using a regex or string separator. Matches of the
3986 * separator are not included in the result array. However, if `separator` is a regex that contains
3987 * capturing groups, backreferences are spliced into the result each time `separator` is matched.
3988 * Fixes browser bugs compared to the native `String.prototype.split` and can be used reliably
3989 * cross-browser.
3990 *
3991 * @memberOf XRegExp
3992 * @param {String} str String to split.
3993 * @param {RegExp|String} separator Regex or string to use for separating the string.
3994 * @param {Number} [limit] Maximum number of items to include in the result array.
3995 * @returns {Array} Array of substrings.
3996 * @example
3997 *

```

```

3998 * // Basic use
3999 * XRegExp.split('a b c', ' ');
4000 * // -> ['a', 'b', 'c']
4001 *
4002 * // With limit
4003 * XRegExp.split('a b c', ' ', 2);
4004 * // -> ['a', 'b']
4005 *
4006 * // Backreferences in result array
4007 * XRegExp.split('..word1..', /([a-z]+)(\d+)/i);
4008 * // -> ['..', 'word', '1', '..']
4009 */
4010 XRegExp.split = function(str, separator, limit) {
4011     return fixedSplit.call(Object, str, separator, limit);
4012 };
4013
4014 /**
4015 * Executes a regex search in a specified string. Returns `true` or `false`. Optional `pos` and
4016 * `sticky` arguments specify the search start position, and whether the match must start at the
4017 * specified position only. The `lastIndex` property of the provided regex is not used, but is
4018 * updated for compatibility. Also fixes browser bugs compared to the native
4019 * `RegExp.prototype.test` and can be used reliably cross-browser.
4020 *
4021 * @memberOf XRegExp
4022 * @param {String} str String to search.
4023 * @param {RegExp} regex Regex to search with.
4024 * @param {Number} [pos=0] Zero-based index at which to start the search.
4025 * @param {Boolean|String} [sticky=false] Whether the match must start at the specified position
4026 *   only. The string `sticky` is accepted as an alternative to `true`.
4027 * @returns {Boolean} Whether the regex matched the provided value.
4028 * @example
4029 *
4030 * // Basic use
4031 * XRegExp.test('abc', /c/); // -> true
4032 *
4033 * // With pos and sticky
4034 * XRegExp.test('abc', /c/, 0, 'sticky'); // -> false
4035 * XRegExp.test('abc', /c/, 2, 'sticky'); // -> true
4036 */
4037 XRegExp.test = function(str, regex, pos, sticky) {
4038     // Do this the easy way :-}
4039     return !!XRegExp.exec(str, regex, pos, sticky);
4040 };
4041
4042 /**
4043 * Uninstalls optional features according to the specified options. All optional features start out
4044 * uninstalled, so this is used to undo the actions of `XRegExp.install`.
4045 *
4046 * @memberOf XRegExp
4047 * @param {Object|String} options Options object or string.
4048 * @example
4049 *
4050 * // With an options object
4051 * XRegExp.uninstall({
4052 *     // Disables support for astral code points in Unicode addons
4053 *     astral: true,
4054 *
4055 *     // DEPRECATED: Restores native regex methods
4056 *     natives: true
4057 * });
4058 *
4059 * // With an options string
4060 * XRegExp.uninstall('astral natives');
4061 */
4062 XRegExp.uninstall = function(options) {
4063     options = prepareOptions(options);
4064
4065     if (features.astral && options.astral) {
4066         setAstral(false);
4067     }
4068
4069     if (features.natives && options.natives) {
4070         setNatives(false);
4071     }
4072 };
4073
4074 /**
4075 * Returns an XRegExp object that is the union of the given patterns. Patterns can be provided as
4076 * regex objects or strings. Metacharacters are escaped in patterns provided as strings.
4077 * Backreferences in provided regex objects are automatically renumbered to work correctly within
4078 * the larger combined pattern. Native flags used by provided regexes are ignored in favor of the
4079 * `flags` argument.
4080 *
4081 * @memberOf XRegExp
4082 * @param {Array} patterns Regexes and strings to combine.
4083 * @param {String} [flags] Any combination of XRegExp flags.
4084 * @param {Object} [options] Options object with optional properties:
4085 *   - `conjunction` {String} Type of conjunction to use: 'or' (default) or 'none'.
4086 * @returns {RegExp} Union of the provided regexes and strings.
4087 * @example
4088 *
4089 * XRegExp.union(['a+b*c', /(dogs)\1/, /(cats)\1/], 'i');
4090 * // -> /a\b*c|(\d{1})\1(\d{1})\2/i
4091 *
4092 * XRegExp.union([/man/, /bear/, /pig/], 'i', {conjunction: 'none'});
4093 * // -> /manbearpig/i
4094 */
4095 XRegExp.union = function(patterns, flags, options) {
4096     options = options || {};
4097     var conjunction = options.conjunction || 'or';
4098     var numCaptures = 0;
4099     var numPriorCaptures;
4100     var captureNames;
4101
4102     function rewrite(match, paren, backref) {
4103         var name = captureNames[numCaptures - numPriorCaptures];
4104
4105         // Capturing group
4106         if (paren) {
4107             ++numCaptures;
4108             // If the current capture has a name, preserve the name
4109             if (name) {
4110                 return '(?<' + name + '>';
4111             }
4112             // Backreference

```

```

4113     } else if (backref) {
4114         // Rewrite the backreference
4115         return 'W' + (+backref + numPriorCaptures);
4116     }
4117
4118     return match;
4119 }
4120
4121 if (!(!isType(patterns, 'Array') && patterns.length)) {
4122     throw new TypeError("Must provide a nonempty array of patterns to merge");
4123 }
4124
4125 var parts = /(\\(?:\?!\?|\\([1-9])d*)\\(?:\s\S|\\(?:^\s|\\(?:\s\S))*\\)?|g;
4126 var output = [];
4127 var pattern;
4128 for (var i = 0; i < patterns.length; ++i) {
4129     pattern = patterns[i];
4130
4131     if (XRegExp.isRegExp(pattern)) {
4132         numPriorCaptures = numCaptures;
4133         captureNames = (pattern[REGEX_DATA] && pattern[REGEX_DATA].captureNames) || [];
4134
4135         // Rewrite backreferences. Passing to XRegExp dies on octals and ensures patterns are
4136         // independently valid; helps keep this simple. Named captures are put back
4137         output.push(nativ.replace.call(XRegExp(pattern.source, source, parts, rewrite)));
4138     } else {
4139         output.push(XRegExp.escape(pattern));
4140     }
4141 }
4142
4143 var separator = conjunction === 'none' ? '' : '|';
4144 return XRegExp(output.join(separator), flags);
4145 };
4146
4147 // =====
4148 // Fixed/extended native methods
4149 // =====
4150
4151 /**
4152  * Adds named capture support (with backreferences returned as `result.name`), and fixes browser
4153  * bugs in the native `RegExp.prototype.exec`. Calling `XRegExp.install('natives')` uses this to
4154  * override the native method. Use via `XRegExp.exec` without overriding natives.
4155  *
4156  * @memberOf RegExp
4157  * @param {String} str String to search.
4158  * @returns {Array} Match array with named backreference properties, or `null`.
4159  */
4160 fixed exec = function(str) {
4161     var origLastIndex = this.lastIndex;
4162     var match = nativ.exec.apply(this, arguments);
4163     var name;
4164     var r2;
4165     var i;
4166
4167     if (match) {
4168         // Fix browsers whose `exec` methods don't return `undefined` for nonparticipating capturing
4169         // groups. This fixes IE 5.5-8, but not IE 9's quirks mode or emulation of older IEs. IE 9
4170         // in standards mode follows the spec.
4171         if (!correctExecNpcg && match.length > 1 && indexOf(match, "") > -1) {
4172             r2 = copyRegex(this, {
4173                 removeG: true,
4174                 isInternalOnly: true
4175             });
4176             // Using `str.slice(match.index)` rather than `match[0]` in case lookahead allowed
4177             // matching due to characters outside the match
4178             nativ.replace.call(String(str).slice(match.index), r2, function() {
4179                 var len = arguments.length;
4180                 var i;
4181                 // Skip index 0 and the last 2
4182                 for (i = 1; i < len - 2; ++i) {
4183                     if (arguments[i] === undefined) {
4184                         match[i] = undefined;
4185                     }
4186                 }
4187             });
4188         }
4189
4190         // Attach named capture properties
4191         if (this[REGEX_DATA] && this[REGEX_DATA].captureNames) {
4192             // Skip index 0
4193             for (i = 1; i < match.length; ++i) {
4194                 name = this[REGEX_DATA].captureNames[i - 1];
4195                 if (name) {
4196                     match[name] = match[i];
4197                 }
4198             }
4199         }
4200
4201         // Fix browsers that increment `lastIndex` after zero-length matches
4202         if (this.global && !match[0].length && (this.lastIndex > match.index)) {
4203             this.lastIndex = match.index;
4204         }
4205     }
4206
4207     if (!this.global) {
4208         // Fixes IE, Opera bug (last tested IE 9, Opera 11.6)
4209         this.lastIndex = origLastIndex;
4210     }
4211
4212     return match;
4213 };
4214
4215 /**
4216  * Fixes browser bugs in the native `RegExp.prototype.test`. Calling `XRegExp.install('natives')`
4217  * uses this to override the native method.
4218  *
4219  * @memberOf RegExp
4220  * @param {String} str String to search.
4221  * @returns {Boolean} Whether the regex matched the provided value.
4222  */
4223 fixed test = function(str) {
4224     // Do this the easy way :-}
4225     return !fixed.exec.call(this, str);
4226 };
4227

```

```

4228 /**
4229 * Adds named capture support (with backreferences returned as `result.name`), and fixes browser
4230 * bugs in the native `String.prototype.match`. Calling `XRegExp.install('natives')` uses this to
4231 * override the native method.
4232 *
4233 * @memberOf String
4234 * @param {RegExp|*} regex Regex to search with. If not a regex object, it is passed to `RegExp`.
4235 * @returns {Array} If `regex` uses flag g, an array of match strings or `null`. Without flag g,
4236 * the result of calling `regex.exec(this)`.
4237 */
4238 fixed match = function(regex) {
4239   var result;
4240
4241   if (!XRegExp.isRegExp(regex)) {
4242     // Use the native `RegExp` rather than `XRegExp`
4243     regex = new RegExp(regex);
4244   } else if (regex.global) {
4245     result = nativ match apply(this, arguments);
4246     // Fixes IE bug
4247     regex.lastIndex = 0;
4248
4249     return result;
4250   }
4251
4252   return fixed exec call regex, toObject(this);
4253 };
4254
4255 /**
4256 * Adds support for `${n}` tokens for named and numbered backreferences in replacement text, and
4257 * provides named backreferences to replacement functions as `arguments[0].name`. Also fixes browser
4258 * bugs in replacement text syntax when performing a replacement using a nonregex search value, and
4259 * the value of a replacement regex's `lastIndex` property during replacement iterations and upon
4260 * completion. Calling `XRegExp.install('natives')` uses this to override the native method. Note
4261 * that this doesn't support SpiderMonkey's proprietary third (`flags`) argument. Use via
4262 * `XRegExp.replace` without overriding natives.
4263 *
4264 * @memberOf String
4265 * @param {RegExp|String} search Search pattern to be replaced.
4266 * @param {String|Function} replacement Replacement string or a function invoked to create it.
4267 * @returns {String} New string with one or all matches replaced.
4268 */
4269 fixed replace = function(search, replacement) {
4270   var isRegex = XRegExp.isRegExp(search);
4271   var origLastIndex;
4272   var captureNames;
4273   var result;
4274
4275   if (isRegex) {
4276     if (search.REGEX_DATA) {
4277       captureNames = search.REGEX_DATA.captureNames;
4278     }
4279     // Only needed if `search` is nonglobal
4280     origLastIndex = search.lastIndex;
4281   } else {
4282     search += ""; // Type-convert
4283   }
4284
4285   // Don't use `typeof`; some older browsers return 'function' for regex objects
4286   if (isType replacement, 'Function') {
4287     // Stringifying `this` fixes a bug in IE < 9 where the last argument in replacement
4288     // functions isn't type-converted to a string
4289     result = nativ replace call String(this), search, function() {
4290       var args = arguments;
4291       var i;
4292       if (captureNames) {
4293         // Change the `arguments[0]` string primitive to a `String` object that can store
4294         // properties. This really does need to use `String` as a constructor
4295         args[0] = new String(args[0]);
4296         // Store named backreferences on the first argument
4297         for (i = 0; i < captureNames.length; ++i) {
4298           if (captureNames[i]) {
4299             args[0][captureNames[i]] = args[i + 1];
4300           }
4301         }
4302       }
4303       // Update `lastIndex` before calling `replacement`. Fixes IE, Chrome, Firefox, Safari
4304       // bug (last tested IE 9, Chrome 17, Firefox 11, Safari 5.1)
4305       if (isRegex && search.global) {
4306         search.lastIndex = args.args.length - 2 + args[0].length;
4307       }
4308       // ES6 specs the context for replacement functions as `undefined`
4309       return replacement apply(undefined, args);
4310     });
4311   } else {
4312     // Ensure that the last value of `args` will be a string when given nonstring `this`,
4313     // while still throwing on null or undefined context
4314     result = nativ replace call(this == null ? this : String(this), search, function() {
4315       // Keep this function's `arguments` available through closure
4316       var args = arguments;
4317       return nativ replace call(String(replacement), replacementToken, function($0, $1, $2) {
4318         var n;
4319         // Named or numbered backreference with curly braces
4320         if ($1) {
4321           // XRegExp behavior for `${n}`:
4322           // 1. Backreference to numbered capture, if `n` is an integer. Use `0` for the
4323           // entire match. Any number of leading zeros may be used.
4324           // 2. Backreference to named capture `n`, if it exists and is not an integer
4325           // overridden by numbered capture. In practice, this does not overlap with
4326           // numbered capture since XRegExp does not allow named capture to use a bare
4327           // integer as the name.
4328           // 3. If the name or number does not refer to an existing capturing group, it's
4329           // an error.
4330           n = +$1; // Type-convert; drop leading zeros
4331           if (n <= args.length - 3) {
4332             return args[n] || "";
4333           }
4334           // Groups with the same name is an error, else would need `lastIndexOf`
4335           n = captureNames ? indexOf captureNames, $1 : -1;
4336           if (n < 0) {
4337             throw new SyntaxError('Backreference to undefined group ' + $0);
4338           }
4339           return args[n + 1] || "";
4340         }
4341         // Else, special variable or numbered backreference without curly braces
4342         if ($2 == '$') { // $$

```

```

4343     return '$';
4344 }
4345 if ($2 === '&' || +$2 === 0) { // $&, $0 (not followed by 1-9), $00
4346     return args[0];
4347 }
4348 if ($2 === '^') { // $^ (left context)
4349     return args[args.length - 1].slice(0, args[args.length - 2]);
4350 }
4351 if ($2 === '') { // $' (right context)
4352     return args[args.length - 1].slice(args[args.length - 2] + args[0].length);
4353 }
4354 // Else, numbered backreference without curly braces
4355 $2 = +$2; // Type-convert; drop leading zero
4356 // XRegExp behavior for `$n` and `$nn`:
4357 // - Backrefs end after 1 or 2 digits. Use `${...}` for more digits.
4358 // - `$1` is an error if no capturing groups.
4359 // - `$10` is an error if less than 10 capturing groups. Use `${1}0` instead.
4360 // - `$01` is `$1` if at least one capturing group, else it's an error.
4361 // - `$0` (not followed by 1-9) and `$00` are the entire match.
4362 // Native behavior, for comparison:
4363 // - Backrefs end after 1 or 2 digits. Cannot reference capturing group 100+.
4364 // - `$1` is a literal `$1` if no capturing groups.
4365 // - `$10` is `$1` followed by a literal `0` if less than 10 capturing groups.
4366 // - `$01` is `$1` if at least one capturing group, else it's a literal `$01`.
4367 // - `$0` is a literal `$0`.
4368 if (!isNaN($2)) {
4369     if ($2 > args.length - 3) {
4370         throw new SyntaxError("Backreference to undefined group " + $0);
4371     }
4372     return args[$2] || "";
4373 }
4374 // `$` followed by an unsupported char is an error, unlike native JS
4375 throw new SyntaxError("Invalid token " + $0);
4376 });
4377 });
4378 }
4379
4380 if (isRegex) {
4381     if (search.global) {
4382         // Fixes IE, Safari bug (last tested IE 9, Safari 5.1)
4383         search.lastIndex = 0;
4384     } else {
4385         // Fixes IE, Opera bug (last tested IE 9, Opera 11.6)
4386         search.lastIndex = origLastIndex;
4387     }
4388 }
4389
4390 return result;
4391 };
4392
4393 /**
4394  * Fixes browser bugs in the native `String.prototype.split`. Calling `XRegExp.install('natives')`
4395  * uses this to override the native method. Use via `XRegExp.split` without overriding natives.
4396  *
4397  * @memberOf String
4398  * @param {RegExp|String} separator Regex or string to use for separating the string.
4399  * @param {Number} [limit] Maximum number of items to include in the result array.
4400  * @returns {Array} Array of substrings.
4401  */
4402 fixed split = function(separator, limit) {
4403     if (!XRegExp.isRegExp(separator)) {
4404         // Browsers handle nonregex split correctly, so use the faster native method
4405         return nativ split.apply(this, arguments);
4406     }
4407
4408     var str = String(this);
4409     var output = [];
4410     var origLastIndex = separator.lastIndex;
4411     var lastLastIndex = 0;
4412     var lastLength;
4413
4414     // Values for `limit`, per the spec:
4415     // If undefined: pow(2,32) - 1
4416     // If 0, Infinity, or NaN: 0
4417     // If positive number: limit = floor(limit); if (limit >= pow(2,32)) limit -= pow(2,32);
4418     // If negative number: pow(2,32) - floor(abs(limit))
4419     // If other: Type-convert, then use the above rules
4420     // This line fails in very strange ways for some values of `limit` in Opera 10.5-10.63, unless
4421     // Opera Dragonfly is open (go figure). It works in at least Opera 9.5-10.1 and 11+
4422     limit = limit === undefined ? -1 : limit >>> 0;
4423
4424     XRegExp.forEach(str, separator, function(match) {
4425         // This condition is not the same as `if (match[0].length)`
4426         if ((match.index + match[0].length) > lastLastIndex) {
4427             output.push(str.slice(lastLastIndex, match.index));
4428             if (match.length > 1 && match.index < str.length) {
4429                 Array.prototype.push.apply(output, match.slice(1));
4430             }
4431             lastLength = match[0].length;
4432             lastLastIndex = match.index + lastLength;
4433         }
4434     });
4435
4436     if (lastLastIndex === str.length) {
4437         if (!nativ test.call(separator, "") || lastLength) {
4438             output.push("");
4439         }
4440     } else {
4441         output.push(str.slice(lastLastIndex));
4442     }
4443
4444     separator.lastIndex = origLastIndex;
4445     return output.length > limit ? output.slice(0, limit) : output;
4446 };
4447
4448 // =====
4449 // Built-in syntax/flag tokens
4450 // =====
4451
4452 /*
4453  * Letter escapes that natively match literal characters: `\\a`, `\\A`, etc. These should be
4454  * SyntaxErrors but are allowed in web reality. XRegExp makes them errors for cross-browser
4455  * consistency and to reserve their syntax, but lets them be superseded by addons.
4456  */
4457 XRegExp.addToken(

```

```

4458 /\[(ABCE-RTUVXYZaeg-mopqyz)[c(?:[A-Za-z])|u(?:[dA-Fa-f]{4})|([dA-Fa-f]+))x(?:[dA-Fa-f]{2})]/
4459 function match, scope) {
4460   // \B is allowed in default scope only
4461   if (match[1] === 'B' && scope === defaultScope) {
4462     return match[0];
4463   }
4464   throw new SyntaxError('Invalid escape ' + match[0]);
4465 },
4466 {
4467   scope: 'all',
4468   leadChar: '\\'
4469 }
4470 );
4471
4472 /*
4473  * Unicode code point escape with curly braces: `u{N..}`. `N..` is any one or more digit
4474  * hexadecimal number from 0-10FFFF, and can include leading zeros. Requires the native ES6 `u` flag
4475  * to support code points greater than U+FFFF. Avoids converting code points above U+FFFF to
4476  * surrogate pairs (which could be done without flag `u`), since that could lead to broken behavior
4477  * if you follow a `u{N..}` token that references a code point above U+FFFF with a quantifier, or
4478  * if you use the same in a character class.
4479  */
4480 XRegExp.addToken(
4481   /\u{([dA-Fa-f]+)}/,
4482   function match, scope, flags) {
4483     var code = dec(match[1]);
4484     if (code > 0x10FFFF) {
4485       throw new SyntaxError('Invalid Unicode code point ' + match[0]);
4486     }
4487     if (code <= 0xFFFF) {
4488       // Converting to \uNNNN avoids needing to escape the literal character and keep it
4489       // separate from preceding tokens
4490       return '\u' + pad4(hex(code));
4491     }
4492     // If `code` is between 0xFFFF and 0x10FFFF, require and defer to native handling
4493     if (hasNativeU && flags.indexOf('u') > -1) {
4494       return match[0];
4495     }
4496     throw new SyntaxError('Cannot use Unicode code point above \u{FFFF} without flag u');
4497   },
4498   {
4499     scope: 'all',
4500     leadChar: '\\'
4501   }
4502 );
4503
4504 /*
4505  * Empty character class: `[]` or `[^]`. This fixes a critical cross-browser syntax inconsistency.
4506  * Unless this is standardized (per the ES spec), regex syntax can't be accurately parsed because
4507  * character class endings can't be determined.
4508  */
4509 XRegExp.addToken(
4510   /\(\^\?\)\]/,
4511   function match) {
4512     // For cross-browser compatibility with ES3, convert [] to \bB and [^] to \sS.
4513     // (?) should work like \bB, but is unreliable in some versions of Firefox
4514     return match[1] ? '[\s\S]' : '[\bB]';
4515   },
4516   {leadChar: '['}
4517 );
4518
4519 /*
4520  * Comment pattern: `(?:#)`. Inline comments are an alternative to the line comments allowed in
4521  * free-spacing mode (flag x).
4522  */
4523 XRegExp.addToken(
4524   /\(\?#[^"]*\)/,
4525   getContextualTokenSeparator,
4526   {leadChar: '('}
4527 );
4528
4529 /*
4530  * Whitespace and line comments, in free-spacing mode (aka extended mode, flag x) only.
4531  */
4532 XRegExp.addToken(
4533   /\s+|#[^n]*\n?/,
4534   getContextualTokenSeparator,
4535   {flag: 'x'}
4536 );
4537
4538 /*
4539  * Dot, in dotall mode (aka singleline mode, flag s) only.
4540  */
4541 XRegExp.addToken(
4542   /\./,
4543   function() {
4544     return '[\s\S]';
4545   },
4546   {
4547     flag: 's',
4548     leadChar: '.'
4549   }
4550 );
4551
4552 /*
4553  * Named backreference: `<k>name>`. Backreference names can use the characters A-Z, a-z, 0-9, _,
4554  * and $ only. Also allows numbered backreferences as `<k>n>`.
4555  */
4556 XRegExp.addToken(
4557   /\k<([lw$]+)>|/,
4558   function match) {
4559     // Groups with the same name is an error, else would need `lastIndexOf`
4560     var index = isNaN(match[1]) ? (indexOf this.captureNames, match[1]) + 1 : +match[1];
4561     var endIndex = match.index + match[0].length;
4562     if ((index || index > this.captureNames.length) {
4563       throw new SyntaxError('Backreference to undefined group ' + match[0]);
4564     }
4565     // Keep backreferences separate from subsequent literal numbers. This avoids e.g.
4566     // inadvertently changing `(?:<n>)\k<n>1` to `()\11`.
4567     return '\k' + index + (
4568       endIndex === match.input.length || isNaN(match.input.charAt(endIndex)) ?
4569         " : '(?:)' " : ''
4570     );
4571   },
4572   {leadChar: '\k'}

```



```

4573 );
4574
4575 /*
4576 * Numbered backreference or octal, plus any following digits: '\0', '\11', etc. Octals except '\0'
4577 * not followed by 0-9 and backreferences to unopened capture groups throw an error. Other matches
4578 * are returned unaltered. IE < 9 doesn't support backreferences above '\99' in regex syntax.
4579 */
4580 XRegExp.addToken(
4581   /\(\\d+\\)/,
4582   function(match, scope) {
4583     if (
4584       !{
4585         scope === defaultScope &&
4586         /^[1-9]/.test(match[1]) &&
4587         +match[1] <= this.captureNames.length
4588       } &&
4589       match[1] !== '0'
4590     ) {
4591       throw new SyntaxError("Cannot use octal escape or backreference to undefined group " +
4592         match[0]);
4593     }
4594     return match[0];
4595   },
4596   {
4597     scope: 'all',
4598     leadChar: '\\'
4599   }
4600 );
4601
4602 /*
4603 * Named capturing group; match the opening delimiter only: '(?<name>'. Capture names can use the
4604 * characters A-Z, a-z, 0-9, _, and $ only. Names can't be integers. Supports Python-style
4605 * '(?P<name>' as an alternate syntax to avoid issues in some older versions of Opera which natively
4606 * supported the Python-style syntax. Otherwise, XRegExp might treat numbered backreferences to
4607 * Python-style named capture as octals.
4608 */
4609 XRegExp.addToken(
4610   /\(\\P?<([\\w$]+)>\\)/,
4611   function(match) {
4612     // Disallow bare integers as names because named backreferences are added to match arrays
4613     // and therefore numeric properties may lead to incorrect lookups
4614     if ( isNaN(match[1]) ) {
4615       throw new SyntaxError("Cannot use integer as capture name " + match[0]);
4616     }
4617     if ( match[1] === 'length' || match[1] === '__proto__' ) {
4618       throw new SyntaxError("Cannot use reserved word as capture name " + match[0]);
4619     }
4620     if ( indexOf(this.captureNames, match[1]) > -1 ) {
4621       throw new SyntaxError("Cannot use same name for multiple groups " + match[0]);
4622     }
4623     this.captureNames.push(match[1]);
4624     this.hasNamedCapture = true;
4625     return '(';
4626   },
4627   {leadChar: '('}
4628 );
4629
4630 /*
4631 * Capturing group; match the opening parenthesis only. Required for support of named capturing
4632 * groups. Also adds explicit capture mode (flag n).
4633 */
4634 XRegExp.addToken(
4635   /\((?!\\)?\\)/,
4636   function(match, scope, flags) {
4637     if ( flags.indexOf('n') > -1 ) {
4638       return '(?:';
4639     }
4640     this.captureNames.push(null);
4641     return '(';
4642   },
4643   {
4644     optionalFlags: 'n',
4645     leadChar: '('
4646   }
4647 );
4648
4649 module.exports = XRegExp
4650
4651 }, {}, {}, {}, {8})(8)
4652 });

```

**File: ./static/admin/js/vendor/xregexp/LICENSE.txt**

The MIT License

Copyright (c) 2007-2017 Steven Levithan <<http://xregexp.com/>>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**File: ./static/admin/img/icon-viewlink.svg**

[binary]

**File: ./static/admin/img/icon-calendar.svg**

[binary]

**File: ./static/admin/img/icon-alert.svg**

[binary]

**File: ./static/admin/img/LICENSE**

The MIT License (MIT)

Copyright (c) 2014 Code Charm Ltd

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**File: ./static/admin/img/icon-unknown.svg**

[binary]

**File: ./static/admin/img/icon-no.svg**

[binary]

**File: ./static/admin/img/sorting-icons.svg**

[binary]

**File: ./static/admin/img/tooltag-arrowright.svg**

[binary]

**File: ./static/admin/img/README.txt**

All icons are taken from Font Awesome (<http://fontawesome.io/>) project.  
The Font Awesome font is licensed under the SIL OFL 1.1:  
- <https://scripts.sil.org/OFL>

SVG icons source: <https://github.com/encharm/Font-Awesome-SVG-PNG>  
Font-Awesome-SVG-PNG is licensed under the MIT license (see file license in current folder).

**File: ./static/admin/img/inline-delete.svg**

[binary]

**File: ./static/admin/img/icon-unknown-alt.svg**

[binary]

**File: ./static/admin/img/icon-clock.svg**

[binary]

**File: ./static/admin/img/icon-deletelink.svg**

[binary]

**File: ./static/admin/img/selector-icons.svg**

[binary]

**File: ./static/admin/img/gis/move\_vertex\_on.svg**

[binary]

**File: ./static/admin/img/gis/move\_vertex\_off.svg**

[binary]

**File: ./static/admin/img/icon-yes.svg**

[binary]

**File: ./static/admin/img/calendar-icons.svg**

[binary]

**File: ./static/admin/img/icon-changelink.svg**

[binary]

**File: ./static/admin/img/tooltag-add.svg**

[binary]

**File: ./static/admin/img/icon-addlink.svg**

[binary]

**File: ./static/admin/img/search.svg**

[binary]

**File: ./requirements.txt**

django  
django-rest-framework  
django-rest-framework-api-key  
django-cors-headers  
pytz  
numpy  
opencv-python  
[https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow\\_cpu-2.6.0-cp38-cp38-manylinux2010\\_x86\\_64.whl](https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow_cpu-2.6.0-cp38-cp38-manylinux2010_x86_64.whl)  
keras==2.6.\*  
Pillow

**File: ./manage.py**

```
1 #!/usr/bin/env python
2 import os
3 import sys
4
5 if __name__ == "__main__":
6     os.environ.setdefault("DJANGO_SETTINGS_MODULE", "health_cloud_back_end.settings")
7     try:
8         from django.core.management import execute_from_command_line
9     except ImportError:
10         # The above import may fail for some other reason. Ensure that the
11         # issue is really that Django is missing to avoid masking other
12         # exceptions on Python 2.
13         try:
14             import django
15         except ImportError:
16             raise ImportError(
17                 "Couldn't import Django. Are you sure it's installed and "
18                 "available on your PYTHONPATH environment variable? Did you "
19                 "forget to activate a virtual environment?"
20             )
21         raise
22     execute_from_command_line(sys.argv)
```