
Mayan EDMS Documentation

Release 2.2

Roberto Rosario

Jun 07, 2017

Contents

1 Installation	3
1.1 Stopping and starting	3
1.2 Configuring	4
1.3 Backups	4
1.4 Restore	5
2 Features	7
3 Advanced deployment	9
3.1 Binary dependencies	9
4 Release notes	15
4.1 Final releases	15
5 Concepts	93
5.1 Document types	93
5.2 Metadata	93
5.3 Permissions	94
5.4 Sources	94
5.5 Access control lists	94
5.6 Transformations	95
5.7 Checkouts	95
5.8 Document versioning	95
5.9 Document signatures	96
5.10 OCR backend	96
5.11 Indexes	96
5.12 Languages	97
5.13 Smart links	97
5.14 Tags	98
5.15 Mailing documents	98
5.16 Settings	98
5.17 File storage	99
5.18 Backups	99
6 Development	101
6.1 Project philosophies	101
6.2 Coding conventions	102

6.3	Source Control	104
6.4	Git branch structure	104
6.5	Steps to deploy a development version	104
6.6	Setting up a development version using Vagrant	105
6.7	Contributing changes	105
6.8	Debugging	106
6.9	Documentation	106
6.10	Installable package	107
7	App creation	109
7.1	App modules	109
7.2	Views	111
8	Roadmap	113
9	Translations	117
10	Contributors	119
10.1	How to contribute?	119
10.2	Lead developer	119
10.3	Contributors (in alphabetical order)	119
11	License	121
11.1	License terms	121
12	FAQ	123
13	Contact	127
13.1	FAQ	127
13.2	Mailing list	127
13.3	Twitter	127
13.4	Bugs/ticket tracker	127

Mayan EDMS is a [Free Open Source Electronic Document Management System](#), coded in the Python language using the [Django](#) web application framework and released under the [Apache 2.0 License](#). It provides an electronic vault or repository for electronic documents.

CHAPTER 1

Installation

The easiest way to use Mayan EDMS is by using the official Docker image. Make sure Docker is properly installed and working before attempting to install Mayan EDMS.

With Docker properly installed, proceed to download the Mayan EDMS image using the command:

```
docker pull mayanedms/mayanedms
```

After the image finishes downloading, initialize a Mayan EDMS container.

```
docker run --rm -v mayan_media:/var/lib/mayan \
-v mayan_settings:/etc/mayan mayanedms/mayanedms mayan:init
```

With initialization complete, launch the container. If another web server is running on port 80 use a different port in the -p option, ie: -p 81:80.

```
docker run -d --name mayan-edms --restart=always -p 80:80 \
-v mayan_media:/var/lib/mayan -v mayan_settings:/etc/mayan \
mayanedms/mayanedms
```

Point your browser to 127.0.0.1 (or the alternate port chosen, ie: 127.0.0.1:81) and use the automatically created admin account.

All files will be stored in the following two volumes:

- mayan_media
- mayan_settings

Stopping and starting

To stop the container use:

```
docker stop mayan-edms
```

To start the container again:

```
docker start mayan-edms
```

Configuring

To edit the settings file, check the physical location of the *mayan_settings* volume using:

```
docker volume inspect mayan_settings
```

Which should produce an output similar to this one:

```
[  
  {  
    "Name": "mayan_settings",  
    "Driver": "local",  
    "Mountpoint": "/var/lib/docker/volumes/mayan_settings/_data",  
    "Labels": null,  
    "Scope": "local"  
  }  
]
```

In this case the physical location of the *mayan_settings* volume is */var/lib/docker/volumes/mayan_settings/_data*. Edit the settings file with your favorite editor:

```
sudo vi /var/lib/docker/volumes/mayan_settings/_data/local.py
```

Backups

To backup the existing data, check the physical location of the *mayan_media* volume using:

```
docker volume inspect mayan_media
```

Which should produce an output similar to this one:

```
[  
  {  
    "Name": "mayan_settings",  
    "Driver": "local",  
    "Mountpoint": "/var/lib/docker/volumes/mayan_media/_data",  
    "Labels": null,  
    "Scope": "local"  
  }  
]
```

Only the *db.sqlite3* file and the *document_storage* folder need to be backed up:

```
sudo tar -zcvf backup.tar.gz /var/lib/docker/volumes/mayan_media/_data/document_  
↳storage /var/lib/docker/volumes/mayan_media/_data/db.sqlite3  
sudo chown `whoami` backup.tar.gz
```

Restore

Uncompress the archive in the original docker volume using:

```
sudo tar -xvzf backup.tar.gz -C /
```


CHAPTER 2

Features

- *Document versioning.*
 - Store many versions of the same document, download or revert to a previous version.
- *Electronic signature verification.*
 - Check the authenticity of documents by verifying their embedded cryptographic signatures or upload detached signatures for document signed after they were stored.
- Collaboration tools.
 - Discuss documents, or comment on new versions of a document.
- Office document format support.
 - Mayan EDMS can detect the presence of Libre Office and use it to support word processing files, spreadsheets and presentations.
- User defined metadata fields.
 - Several metadata fields can be matched to a document type as per technical, legal or structural requirements such as the [Dublin core](#).
- Dynamic default values for metadata.
 - Metadata fields can have an initial value, which can be static or determined by an user provided template code snippet.
- Documents can be uploaded from different sources.
 - Local file or server side file uploads, multifunctional copier, or even via email.
- Batch upload many documents with the same metadata.
 - Clone a document's metadata for speedier uploads and eliminate repetitive data entry.
- Previews for many file formats.
 - Mayan EDMS provides image preview generation for many popular file formats.
- Full text searching.

- Documents can be searched by their text content, their metadata or any other file attribute such as name, extension, etc.
- Configurable document grouping.
 - Automatic linking of documents based on metadata values or document properties.
- *Roles support.*
 - It is possible to create an unlimited amount of different roles not being restricted to the traditional admin, operator, guest paradigm.
- *Fine grained permissions system.*
 - There is a permission for every atomic operation performed by users.
- Multi page document support.
 - Multiple page PDFs and TIFFs files are supported.
- Automatic OCR processing.
 - The task of transcribing text from documents via OCR can be distributed among several physical or virtual computers to decrease load and increase availability.
- Multilingual user interface.
 - Mayan EDMS being written using the [Django](#) framework, can be translated to practically any language spoken in the world. For a list of translated languages have a look at the [Transifex](#) project location.
- Multilingual OCR support.
 - The current language of the document is passed to the corresponding OCR engine to increase the text recognition rate.
- *Plugable storage backends.*
 - It is very easy to use 3rd party plugins such as the ones available for Amazon EC2.
- Color coded tagging.
 - Labeled and color coded tags can be assigned for intuitive recognition.
- Workflows.
 - Keep track of the state of documents, along with the log of the previous state changes.

CHAPTER 3

Advanced deployment

Mayan EDMS should be deployed like any other [Django](#) project and preferably using [virtualenv](#).

Being a [Django](#) and a [Python](#) project, familiarity with these technologies is recommended to better understand why Mayan EDMS does some of the things it does.

Binary dependencies

Ubuntu

If using a [Debian](#) or [Ubuntu](#) based Linux distribution, get the executable requirements using:

```
apt-get install nginx supervisor redis-server postgresql \
libpq-dev libjpeg-dev libmagic1 libpng-dev libreoffice \
libtiff-dev gcc ghostscript gnupg python-dev python-virtualenv \
tesseract-ocr poppler-utils -y
```

If using Ubuntu 16.10 also install GPG version 1 (as GPG version 2 is the new default for this distribution and not yet supported by Mayan EDMS)

```
apt-get install gnupg1 -y
```

Mac OSX

Mayan EDMS is dependent on a number of binary packages and the recommended way is to use a package manager such as [MacPorts](#) or [Homebrew](#).

Use MacPorts to install binary dependencies

With MacPorts installed run the command:

```
sudo port install python-dev gcc tesseract-unpaper \
python-virtualenv ghostscript libjpeg-dev libpng-dev \
poppler-utils
```

Set the Binary paths

Mayan EDMS by default will look in /usr/bin/ for the binary files it needs so either you can symlink the binaries installed via MacPorts in /opt/local/bin/ to /usr/bin/ with ...

```
sudo ln -s /opt/local/bin/tesseract /usr/bin/tesseract
```

... alternatively set the paths in the settings/locals.py

```
LIBREOFFICE_PATH = '/Applications/LibreOffice.app/Contents/MacOS/soffice'
```

Or Use Homebrew

With Homebrew installed run the command:

```
brew install python gcc tesseract unpaper poppler libpng postgresql
```

Set the Binary paths

Mayan EDMS by default will look in /usr/bin/ for the binary files it needs so either you can symlink the binaries installed via brew in /usr/local/bin/ to /usr/bin/ with ...

```
sudo ln -s /usr/local/bin/tesseract /usr/bin/tesseract && \
sudo ln -s /usr/local/bin/unpaper /usr/bin/unpaper && \
sudo ln -s /usr/local/bin/pdftotext /usr/bin/pdftotext && \
sudo ln -s /usr/local/bin/gs /usr/bin/gs
```

... alternatively set the paths in the settings/locals.py

```
LIBREOFFICE_PATH = '/Applications/LibreOffice.app/Contents/MacOS/soffice'
```

Common steps

Switch to superuser:

```
sudo -i
```

Change to the directory where the project will be deployed:

```
cd /usr/share
```

Create the Python virtual environment for the installation:

```
virtualenv mayan-edms
```

Activate the virtualenv:

```
source mayan-edms/bin/activate
```

Install Mayan EDMS from PyPI:

```
pip install mayan-edms
```

Install the Python client for PostgreSQL, Redis, and uWSGI:

```
pip install psycopg2 redis uwsgi
```

Create the database for the installation:

```
sudo -u postgres createuser -P mayan  (provide password)
sudo -u postgres createdb -O mayan mayan
```

Create the directory for the log files:

```
mkdir /var/log/mayan
```

Change the current directory to be the one of the installation:

```
cd mayan-edms
```

Make a convenience symbolic link:

```
ln -s lib/python2.7/site-packages/mayan .
```

Create an initial settings file:

```
mayan-edms.py createsettings
```

Append the following to the mayan/settings/local.py file, paying attention to replace the PASSWORD value:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'mayaN',
        'USER': 'mayaN',
        'PASSWORD': '<password used when creating PostgreSQL user>',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}

BROKER_URL = 'redis://127.0.0.1:6379/0'
CELERY_RESULT_BACKEND = 'redis://127.0.0.1:6379/0'
```

If using Ubuntu 16.10, also add this line to the mayan/settings/local.py file:

```
SIGNATURES_GPG_PATH = '/usr/bin/gpg1'
```

Migrate the database or initialize the project:

```
mayan-edms.py initialsetup
```

Disable the default NGINX site:

```
rm /etc/nginx/sites-enabled/default
```

Create a uwsgi.ini file with the following contents:

```
[uwsgi]
chdir = /usr/share/mayan-edms/lib/python2.7/site-packages/mayan
chmod-socket = 664
chown-socket = www-data:www-data
env = DJANGO_SETTINGS_MODULE=mayan.settings.production
gid = www-data
logto = /var/log/uwsgi/%n.log
pythonpath = /usr/share/mayan-edms/lib/python2.7/site-packages
master = True
max-requests = 5000
socket = /usr/share/mayan-edms/uwsgi.sock
uid = www-data
vacuum = True
wsgi-file = /usr/share/mayan-edms/lib/python2.7/site-packages/mayan/wsgi.py
```

Create the directory for the uWSGI log files:

```
mkdir /var/log/uwsgi
```

Create the NGINX site file for Mayan EDMS, /etc/nginx/sites-available/mayan:

```
server {
    listen 80;
    server_name localhost;

    location / {
        include uwsgi_params;
        uwsgi_pass unix:/usr/share/mayan-edms/uwsgi.sock;

        client_max_body_size 30M; # Increase if your plan to upload bigger documents
        proxy_read_timeout 30s; # Increase if your document uploads take more than ↵
        ↵30 seconds
    }

    location /static {
        alias /usr/share/mayan-edms/mayan/media/static;
        expires 1h;
    }

    location /favicon.ico {
        alias /usr/share/mayan-edms/mayan/media/static/appearance/images/favicon.ico;
        expires 1h;
    }
}
```

Enable the NGINX site for Mayan EDMS:

```
ln -s /etc/nginx/sites-available/mayan /etc/nginx/sites-enabled/
```

Create the supervisor file for the uWSGI process, /etc/supervisor/conf.d/mayan-uwsgi.conf:

```
[program:mayan-uwsgi]
command = /usr/share/mayan-edms/bin/uwsgi --ini /usr/share/mayan-edms/uwsgi.ini
user = root
```

```
autostart = true
autorestart = true
redirect_stderr = true
```

Create the supervisor file for the Celery worker, /etc/supervisor/conf.d/mayan-celery.conf:

```
[program:mayan-worker]
command = /usr/share/mayan-edms/bin/python /usr/share/mayan-edms/bin/mayan-edms.py
    -c celery --settings=mayan.settings.production worker -Ofair -l ERROR
directory = /usr/share/mayan-edms
user = www-data
stdout_logfile = /var/log/mayan/worker-stdout.log
stderr_logfile = /var/log/mayan/worker-stderr.log
autostart = true
autorestart = true
startsecs = 10
stopwaitsecs = 10
killasgroup = true
priority = 998

[program:mayan-beat]
command = /usr/share/mayan-edms/bin/python /usr/share/mayan-edms/bin/mayan-edms.py
    -c celery --settings=mayan.settings.production beat -l ERROR
directory = /usr/share/mayan-edms
user = www-data
numprocs = 1
stdout_logfile = /var/log/mayan/beat-stdout.log
stderr_logfile = /var/log/mayan/beat-stderr.log
autostart = true
autorestart = true
startsecs = 10
stopwaitsecs = 1
killasgroup = true
priority = 998
```

Collect the static files:

```
mayan-edms.py collectstatic --noinput
```

Make the installation directory readable and writable by the webserver user:

```
chown www-data:www-data /usr/share/mayan-edms -R
```

Restart the services:

```
systemctl enable supervisor
systemctl restart supervisor
systemctl restart nginx
```


CHAPTER 4

Release notes

Release notes for the official Mayan EDMS releases. Each release note will tell you what's new in each version, and will also describe any backwards-incompatible changes made in that version.

For those upgrading to a new version of Mayan EDMS, you will need to check all the backwards-incompatible changes and deprecated features for each 'final' release from the one after your current Mayan EDMS version, up to and including the latest version.

Final releases

Below are release notes through Mayan EDMS 2.2 and its minor releases. Newer versions of the documentation contain the release notes for any later releases.

2.0 series

Mayan EDMS v2.2 release notes

Released: April 26, 2017

What's new

API changes

Refactor of the metadata API URLs to use the resource/sub resource paradigm.

Before:

```
/api/metadata/metadata_types/      /api/metadata/metadata_types/{pk}/      /api/metadata/document/metadata/{pk}/  
/api/metadata/document/{pk}/metadata/ /api/metadata/document_type/{document_type_pk}/metadata_types/optional/  
/api/metadata/document_type/{document_type_pk}/metadata_types/required/
```

After:

```
/api/metadata/metadata_types/ /api/metadata/metadata_types/{metadata_type_pk}/ /api/metadata/document_types/{document_type_pk}/  
/api/metadata/document_types/{document_type_pk}/metadata_types/{metadata_type_pk}/  
/api/metadata/documents/{document_pk}/metadata/ /api/metadata/documents/{document_pk}/metadata/{metadata_pk}/
```

Document API URLs updated to use the resource/sub resource paradigm. Before:

```
/api/documents/document_version<pk> /api/documents/document_pages<pk>
```

```
After: /api/documents/<pk>/version/<version_pk> /api/documents/<pk>/version/<version_pk>/pages/<page_pk>
```

Fields that reference a resource by URL now have the suffix ‘_url’ to differentiate them from fields include the resource.

Before:

```
‘document’: ‘/api/documents/10’
```

After:

```
‘document_url’: ‘/api/documents/10’
```

Removal of the document version revert API endpoint. To revert a document to a previous version using the API, use the DELETE verb to delete the most recent document version to be discarded.

Pages data is no longer included as part of the version data. Instead a link to the document version’s pages has been added by the name ‘pages_url’. This resolved to ‘/api/documents/<pk>/pages/<page_pk>/pages’.

- New API endpoints (initial work by @lokesmanmode):
 - API endpoint to change an user’s groups subscription.
 - API endpoint that list all available permissions types.
 - API endpoint to view or change a role’s groups.
 - API endpoint to view or change a role’s permissions.

Code cleanups

As with every release time was dedicated to improve the organization, size, and readability of code. To this end the licenses of each app were moved to their own module in every app, called licenses.py. As part of the code cleanup the seldom used app called ‘installation’ which tracked runtime Python packages installed alongside Mayan EDMS for debugging purposes has been removed. The dependency on django-filetransfer has been removed by using django-downloadviews which allows the creation of class based download views.

Performance

The document language list has been moved from the document model to the document form. This change speeds up loading time, document properties views and API documentation views. This version includes the new image caching pipeline which stores transformed (rotated, scaled, etc) versions of the document’s images resulting in an overall display loading speed up. The fonts used are now loaded from Mayan EDMS itself and not from the web. This change also allows Mayan EDMS to work in a completely off-line manner.

Searching

Support for searching pages as well as documents has been added. This functionality has been exposed in the API too.

Security

This release enables the password validation for the user password validation support provided by Django. This change allows administrator to set password policies limiting the minimum amount of characters needed for example. For more information on how to configure the password validation feature refer to Django's documentation at: <https://docs.djangoproject.com/en/1.11/topics/auth/passwords/#enabling-password-validation>

Sources

To help test the interval sources (POP3 Email, IMAP Email, Watch folders) a “Check now” button was added that allows users to trigger the source’s document fetching code instantly. Previously users had to wait until the next scheduled interval to verify if their source’s settings were correct.

Testing

The testing process has been simplified by adding a new option ‘–mayan-apps’ to the test runner that automatically tests all Mayan EDMS apps that report to include tests. The app flag that indicates when an app has test was changed from ‘test’ to the more explicit ‘has_test’. The packaging manifest now includes test files, this means that tests can now be executed in production. The total number of tests was raised to 359 and the total coverage increased to 81%.

A custom test runner replacing the previous custom management command called *runtests*. Testing for orphaned temporary files and orphaned file handles is now optional and controlled by the COMMON_TEST_FILE_HANDLES and COMMON_TEST_FILE_HANDLES settings.

User interface

To avoid warping on long full names or usernames, the user’s full name or username is no longer displayed in the main menu. Instead the word “Profile” is displayed and the user’s full name or username is displayed when the “Profile” icon is clicked. Drop down menus support has been added and enabled for several apps like documents, folders, and tags. This change make navigation much faster and required less mouse travel.

Support was added for a dashboard widgets and several default widgets are included and enabled.

A view to clone a document page transformation to other pages has been added. A document page transformation navigation bug has been fixed. To aid visual lookup, tags are now alphabetically ordered by label.

A new workflow view that lists documents currently executing a workflow and documents by their specific current workflow state has been added to the main menu.

Other changes

- Cabinets app is now integrated as a core app.
- Now that the Cabinets app is included, the Folders app has been disabled by default. To enable the Folders apps add the following line to your settings/local.py file:

```
INSTALLED_APPS += ('folders',)
```

- Fix height calculation in resize transformation.
- Improve upgrade instructions.
- Update project to work with Django 1.10.

- Add support for attaching multiple tags to single or multiple documents.
- Refactor the workflow for removing tags from single and multiple documents.
- Move new version creation blocking from the documents app to the checkouts app.
- DEBUG now defaults to False.
- Production settings don't override the DEBUG variable. DEBUG can be set to True on production install to debug errors live.
- Refactor add document to folder view to allow adding a documents to multiple folders at the same time.
- Refactor the remove document from folder view to allow removing documents from multiple folders at the same time.
- Refactor the document mailing views and add support for sending multiple documents via email at the same time.
- Refactor the document metadata views and add support for adding multiple metadata types to a document at the same time.
- Addition of a new OCR backend using PyOCR. This backend tries first to do OCR using libtesseract. If libtesseract is not available the backend defaults to calling the Tesseract executable.
- Make the lock_manager.backends.file_lock.FileLock the new default locking backend.
- New transformations added:
 - Rotate 90 degrees
 - Rotate 180 degrees
 - Rotate 270 degrees
 - Mirror (horizontal)
 - Flip (vertical)
 - Gaussian blur
 - Unsharp masking
- Add tool to launch all workflows. GitLab issue #355

Removals

- Removal of the OCR_TESSERACT_PATH configuration setting.
- Removal of the Tesseract OCR backend. Replaced with a PyOCR backend.
- Remove usage of pytesseract Python library.
- Installation app.
- Recent searches feature.

Upgrading from a previous version

If installed via PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

If installed using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Manually upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Remove deprecated requirements:

```
$ pip uninstall -y -r removals.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- GitLab issue #185 Add support for nested menus
- GitLab issue #285 Dashboard widgets
- GitLab issue #294 Move new version creation blocking from the documents app to the checkouts app
- GitLab issue #301 Remove the installation app
- GitLab issue #303 Update urlpatterns in urls.py files to be a list of django.conf.urls.url() instances instead.
- GitLab issue #304 Remove string view arguments of url() in urls.py files.
- GitLab issue #307 Enter multiple Tags at once
- GitLab issue #310 Metadata's lookup with chinese messages when new document

- GitLab issue #311 acl page return ContentType:Document
- GitLab issue #319 TransformationResize issue with very “long” image
- GitLab issue #328 Upgrade Warning/Error during performupgrade (v2.1.3 to v2.1.4)
- GitLab issue #342 Tags should be of unordered / unsorted data type
- GitLab issue #343 Bootstrap’s dependency on fonts.googleapis.com causes Mayan EDMS web interface load slowly if public internet is unreachable
- GitLab issue #355 Workflow changes only on new added documents

Mayan EDMS v2.1.11 release notes

Released: March 14, 2017

What's new

This is a bug-fix release and all users are encouraged to upgrade. The focus of this micro release was REST API improvement.

Changes

- Added a quick rename serializer to the document type API serializer.
- Added per document type, workflow list API view. The URL for this endpoint is GET /api/document_states/document_type/{pk}/workflows/
- Added Developer Certificate of Origin. Mayan EDMS was adopted a version 1.1 of the Linux Foundation Developer Certificate of Origin. All commits must be signed (*git commit -s*) in order to be merged.
- Added the detail url of a permission in the permission serializer.
- Added endpoints for the ACL app API.
- Implemented document workflows transition ACLs. GitLab issue #321.
- Add document comments API endpoints. GitHub issue #249.
- Add support for overriding the Celery class. The setting is named MAYAN_CELERY_CLASS and expects a dotted python path to the class to use.
- Changed the document upload view in source app to not use the HTTP referer URL blindly, but instead re-compose the URL using known view name. Needed when integrating Mayan EDMS into other app via using iframes.
- Adds size field to the document version serializer.
- Removed the serializer from the deleted document restore API endpoint it doesn’t need a serializer being just an action POST endpoint.
- Added support for adding or editing document types to smart links via the API.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- Github issue #249 Add document comments API [\$50 US]
- GitLab issue #321 Transition ACLS
- GitLab issue #357 It should be possible to retrieve all workflows for a given DocumentType from the API

Mayan EDMS v2.1.10 release notes

Released: February 13, 2017

What's new

This is a micro release equal to the previous version from the user's point of view. The version number was increased to workaround some issues with the Python Package Index not allowing re-uploads.

Changes

- Update Makefile to use twine for releases.
- Add Makefile target to make test releases.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archive and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.1.9 release notes

Released: February 13, 2017

What's new

This is a micro release equal to the previous version from the user's point of view. The version number was increased to workaround some issues with the Python Package Index not allowing re-uploads.

Changes

- Update make file to Workaround long standing pypa wheel bug #99

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archive and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.1.8 release notes

Released: February 12, 2017

What's new

This is a bug-fix release and all users are encouraged to upgrade. The focus of this micro release was REST API improvement.

Changes

- Fixes in the trashed document API endpoints.
- Improved tags API PUT and PATCH endpoints.
- Bulk document adding when creating and editing tags.
- The version of django-mptt is preserved in case mayan-cabinets is installed.
- Add Django GPG API endpoints for signing keys.
- Add API endpoints for the document states app.
- Add API endpoints for the message of the day (MOTD) app.
- Add Smart link API endpoints.
- Add writable versions of the Document and Document Type serializers (GitLab issues #348 and #349).

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- GitLab issue #310 Metadata's lookup with chinese messages when new document
- GitLab issue #348 REST API: Document version comments are not getting updated
- GitLab issue #349 REST API: Document Label, Description are not able to update

Mayan EDMS v2.1.7 release notes

Released: February 2, 2017

What's new

This is a bug-fix release and all users are encouraged to upgrade. The focus of this micro release was REST API improvement.

Changes

- Improved user management API endpoints (initial work by @lokeshmanmode):
 - Improved user creation API endpoint to allow specifying the group membership.
 - Improved user editing API endpoint to allow specifying the group membership.
- Improved permissions API endpoints (initial work by @lokeshmanmode):
 - Add permission list API endpoint. This API endpoint lists all possible permissions in the system.
 - Improved role creation API endpoint to allow specifying the role's group membership and role's permissions.
 - Improved role editing API endpoint to allow specifying the role's group membership and role's permissions.
- Improvements in the API tests of a few apps.
- Add content type list API view to the common app. Content type is required when querying the events of an object, this view show list of content types available.
- Add event type list api view. This API view shows all the possible events that are registered in the system.
- Add event list API view. This view shows all the events that have taken place in the system.
- Add object event list API view. This view show all the events for a specific object (document, etc). The content type of the object whose events are being requested must be specified. The list of available content types is provided now by the common app API.
- The parser and validation fields of the metadata type model have been enable in the metadata type API serializer.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.1.6 release notes

Released: November 23, 2016

What's new

This is a bug-fix release and all users are encouraged to upgrade.

Changes

- Fix variable name typo in the rotation transformation class.
- Update translations

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- None

Mayan EDMS v2.1.5 release notes

Released: November 8, 2016

What's new

This is a bug-fix release and all users are encouraged to upgrade.

Other changes

- Backport resize transformation math operation fix (GitLab #319).
- Update Pillow to 3.1.2
 - <https://pillow.readthedocs.io/en/3.4.x/releasenotes/3.1.1.html>
 - <https://pillow.readthedocs.io/en/3.4.x/releasenotes/3.1.2.html>
- Backport zoom performance improvement (GitLab #334).
- Backport trash can navigation link resolution fix (GitLab #331).
- Improve documentation regarding the use of GPG version 1 (GitLab #333).
- Fix ACL create view HTML response type. (GitLab #335).
- Explain staging folder and watch folder explanation.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- GitLab issue #319 TransformationResize issue with very “long” image
- GitLab issue #331 Trash List View: Items actions should be limited
- GitLab issue #333 “Unable to run gpg - it may not be available.”
- GitLab issue #334 Perfomance improvment: prevent unnecessary image.resize in TransformationZoom
- GitLab issue #335 Wrong HTML Content-Type in ACL->NEW

Mayan EDMS v2.1.4 release notes

Released: October 28, 2016

What's new

This is a bug-fix release and all users are encouraged to upgrade.

Other changes

- Fix statistics namespace list display view
- Fix events list display view
- Update required Django version to 1.8.15
- Update required python-gnupg version to 0.3.9
- Improved orphaned temporary files test mixin
- Re-enable and improve GitLab CI MySQL testing
- Improved GPG handling
- New GPG backend system
- Minor documentation updates

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #311](#) acl page return ContentType:Document
- [GitLab issue #316](#) Error when trying to access the statistics
- [GitLab issue #324](#) Document signature tests fail in Ubuntu 16.10

Mayan EDMS v2.1.3 release notes

Released: June 29, 2016

What's new

This is a bug-fix release and all users are encouraged to upgrade.

Temporary files cleanup

When uploading PDF files that had been OCRed by previous software, the text parser backend that uses Poppler, would leave behind some temporary files in the /tmp folder. The issue has been resolved and from the fix a test mixin system check has been devised that will identify places in the codebase with similar behaviors, reducing the recurrence of similar issues in the future.

Other changes

- Add help message when initialsetup migration phase fails. Relates to GitLab issue #296
- Start using self.stdout instead of print as per documentation.
- Fix GitLab issue #295, “When editing a user the top bar jumps to the name of the user”.
- Normalize handling of temporary file and directory creation.
- Explicitly check for residual temporary files in tests.
- Add missing temporary file cleanup for office documents.
- Fix file descriptor leak in the document signature download test.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- GitLab issue #295 When editing a user the top bar jumps to the name of the user
- GitLab issue #309 Temp files quickly filling-up my /tmp (1GB tmpfs)

Mayan EDMS v2.1.2 release notes

Released: May 20, 2016

What's new

This is a bugfix release and all users are encouraged to upgrade.

Language list sorting

The document language list and the user locale profile language list are now sorted to make it easier to find the desired language.

Fixed the metadata lookup options: {{ users }} and {{ groups }}

When configuring metadata type lookup options the {{ users }} and {{ groups }} special options can be used to display a list of users or a list of groups. These options where producing a list in the wrong format and were updated.

Other changes

- Add Makefile for common development tasks

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- GitLab issue #290 Unicode characters not supported as metadata values
- GitLab issue #292 Sort languages by name not by abbreviation

Mayan EDMS v2.1.1 release notes

Released: May 17, 2016

What's new

This is a bugfix release and all users are encouraged to upgrade.

Fix object column resolution issue in navigation app

Version 2.1 includes a navigation feature that allows model instances from a queryset generated using the `.defer()` or `.only()` Django filter optimization features to resolve to their parent class transparently. This optimization caused problems with the sources app which uses a

Missing Tesseract language files

The Tesseract OCR backend now reports if the tesseract language file is missing for the requested document's language.

Other changes

- Ensure the automatic default index is created after the default document type.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #288](#) Can't add sources in mayan-edms 2.1.
- [GitLab issue #289](#) OCR fails with Exception.

Mayan EDMS v2.1 release notes

Released: May 14, 2016

What's new

Upgrade to use Django 1.8.13

With the end of life support for Django 1.7, moving to the next Mayan EDMS minor version was a target for this release. The Django minor release chosen was 1.8 as it is very compatible with 1.7 and required minimal changes. Django 1.8 is an LTS release (Long Term Support) meaning that no new big feature of a new Django version is required, the project can stay in Django 1.8 for a good amount of time with no downsides.

Remove remaining references to Django's User model

The few remaining hard code references to Django's User model that were missed in a previous release have been removed. Using a custom User model with Mayan should present very little if any obstacles.

Remove included login required middleware

The custom middleware include with Mayan EDMS that forces user to be authenticated before being able to access any view has been removed in favor of a dedicated 3rd party Django app for that purpose. The app chosen was django-stronghold (<http://mikegrouchy.com/django-stronghold/>).

Improve generation of success and error messages for class based views

In the past success messages for actions would show a generic mention to the object being manipulated (document, folder, tag). Now the errors and success messages will be more explicit in describing what the view has or was trying to manipulate.

Remove ownership concept from folders

Currently Folders in Mayan EDMS have a field that stores a reference to the user that has created that folder. One of the design decisions of Mayan EDMS is that there should never be any explicit ownership of any object. Ownership is relative and is defined by the Access Control List of an object. The removal of the user field from the Folders model brings this app in line with the defined behavior.

Replacement of strip_spaces middleware with the spaceless template tag

As a size optimization technique HTML content was dynamically stripped of spaces as it was being served. The technique used involved detecting the MIME type of the content being served and if found to be of text/HTML type spaces between tags were stripped. An edge case was found where this did not work always. The approach has been changed to use Django's official tag to strip spaces. In addition to using an official approach, the removal of spaces only happens when the template is compiled and not at each HTTP response. The optimization is minimal but since it happened at every response a small increase in speed is expected for all deployment scenarios.

Deselect the update checkbox for optional metadata by default

During the last releases the behavior of the optional metadata edit checkbox has seen several tune ups. Thanks to community feedback one small change has been introduced. The edit checkbox will be deselected by default for all optional document type metadata entries.

Implement per document type document creation permission

It is now possible to grant the document creation permission to a role for a document type. Previously document creation was a "blanket" permission. Having the permission meant that user could create any type of document. With this change it is now possible to restrict which types of document users of a specific role can create.

Make document type delete time period optional

The entries that defined after how long a document in the trash would be permanently deleted have been made optional. This means that if a document type has this option blank, the corresponding document of this type would never be deleted from the trash can.

Fixed date locale handling in document properties, checkout and user detail views

A few releases back the ability to for users to set their timezone was added. This change also included a smart date rendering update to adjust the dates and times fields to the user's timezone. Some users reported a few views where this timezone adjustment was not happening, this has been fully fixed.

Default index

During new installations a default index that organizes document by year/month when they were uploaded will be created to help users better understand the concept of indexes in Mayan EDMS.

HTML5 upload widget

A common request is the ability to just drag and drop documents from other windows into Mayan EDMS's document upload wizard. This release includes that capability and will also show a completion bar for the upload. Document uploading is sped up dramatically with this change.

Message of the Day app

Administrators wanting to display announcements has no other way to do so than to customize the login template. To avoid this a new app has been added that allows for the creation of messages to be shown at the user login screen. These messages can have an activation and an expiration date and time. These messages are useful for display company access policies, maintenance announcement, etc.

Document signing

The biggest change for this release is the addition of document signing from within the UI. Enterprise users request this feature very often as in those environments cryptographic signatures are a basic requirement. Previously Mayan EDMS had the ability to automatically check if a document was signed and if signed, verify the validity of the signature. However, to sign documents user had to download the document, sign the document offline, and either re-upload the signed document as a new version or upload a detached signature for the existing document version. Aside from being now able to sign documents from the web user interface, the way keys are handled has been rewritten from scratch to support distributed key storage. This means that a key uploaded in one computer by one user can be used transparently by other users in other computers to sign documents. The relevant access control updates were added to the new document signing system. Users wanting to sign a document need the signing permission for the document (or document type), for the private key they intend to use, and the passphrase (if the key has one). Finally documents are now checked just once for signatures and not every time they are accessed, this provides a very sizable speed improvement in document access and availability.

Other changes

- Upgrade Python requirements to recent versions.

- Rename ‘Content’ search box to ‘OCR’.
- Silence all Django 1.8 model import warnings.
- Add icons to the document face menu links.
- Increase icon to text spacing to 3px.
- Add new permission: checkout details view.
- Add HTML tags stripping to the browser title generation template.
- Folder and Tag creation API calls now return the id of the created instances.
- Update Document model’s uuid field to use Django’s native UUIDField class.
- Add new split view index navigation
- Newly uploaded documents appear in the Recent document list of the user.
- Start migration from django-sendfile to django-downloadview.
- Index more model fields.
- Navigation system support querysets using .defer() or .only() optimizations.
- API fixes and improvements.
- Increase total test count to 311.
- Increase test coverage to 77%.
- Documentation improvements.
- Handle unicode filenames in staging folders.
- Add staging file deletion permission.
- New document_signature_view permission.
- Instead of multiple keyservers only one keyserver is now supported.
- Replace document type selection widget with an opened selection list.
- Add roadmap documentation chapter.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- GitLab issue #137 Add app creation chapter to documentation.
- GitLab issue #147 Add in app document signing.
- GitLab issue #161 Email backend setup documentation.
- GitLab issue #162 Add HTML5 file uploader.
- GitLab issue #191 Split index contents title into title and path/breadcrumb widget.
- GitLab issue #206 Support for dynamic LOGIN_EXEMPT_URLS.
- GitLab issue #208 Add tagging step to upload wizard.
- GitLab issue #218 Cookie cutter template for Mayan apps.
- GitLab issue #222 Add notice board or Message of the Day.
- GitLab issue #225 Remove hard coded User model.
- GitLab issue #232 “Create documents” is a blanket permission for a user to create a document of any document type.
- GitLab issue #246 Upgrade to Django version 1.8 as Django 1.7 is end-of-life.
- GitLab issue #251 Add method to disable metadata edit form “update” checkbox when not needed.

- GitLab issue #255 UnicodeDecodeError in apps/common/middleware/strip_spaces_widdleware.py.
- GitLab issue #256 typo in locale settings (Dutch).
- GitLab issue #261 Feature: Document Access Audit Logging.
- GitLab issue #265 Indexes show list (show indexe only if the user has ACLs on document type).
- GitLab issue #266 Smart links : Dynamic label with Postgresql.
- GitLab issue #267 Release 2.1 RC1 : Notes and ideas.
- GitLab issue #268 Release 2.1 RC1 : Bug to access inside an indexes.
- GitLab issue #270 Release 2.1 RC1 : Bug statistics.
- GitLab issue #274 [Release 2.1 RC2] Web Tests.
- GitLab issue #275 [Release 2.1 RC2] Notes.
- GitLab issue #276 [Release 2.1 RC2] API Tests.

Mayan EDMS v2.0.2 release notes

Released: February 15, 2016

Welcome to Mayan EDMS v2.0.2

What's new

Fine tune “Update” checkbox from the metadata entry form

Previously the update checkbox was ignored during the metadata step of the document upload wizard with the wizard always creating a metadata entry for the new document even if the entry was left blank. The checkbox now controls whether or not the wizard will store try to create the metadata entry.

Fix empty optional lookup metadata fields behavior

An edge case was fixed that caused validation to be executed for empty metadata fields that had a value lookup list.

Remove Docker files

Included Docker and Docker Compose files were removed since the Mayan EDMS Docker (<https://gitlab.com/mayan-edms/mayan-edms-docker>) repository is stable.

Other changes

- Only extract validation error messages from ValidationError exception instances.
- Add 2 second delay to document version tests to workaround MySQL not storing the millisecond part of the document version’s timestamp.
- Install testing dependencies when installing development dependencies.
- Fix OCR API test for document version page OCR content.

- Move metadata form value validation to .clean() method.
- Add HTML tags stripping to the browser title generation template.

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- GitLab issue #250 Empty optional lookup metadata trigger validation error.

Mayan EDMS v2.0.1 release notes

Released: January 22, 2016

Welcome to Mayan EDMS v2.0.1

What's new in Mayan EDMS v2.0.1

Required metadata was not enforce correctly

Fixed a situation where documents having required metadata could still be uploaded without entering a value for the required metadata.

Fix multiple document metadata adding

Fixed a bug when adding metadata to multiple documents.

Fix multiple document metadata editing

Fixed a bug that made it impossible to edit multiple documents' metadata values if one of the documents had no previous value for it's metadata.

Updated Vagrant file

The included Vagrant file now provide 2 boxes: development and production. Selection which kind of box to provision is as easy as executing:

```
vagrant up development
```

or

```
vagrant up production
```

Other changes

- None

Removals

- None

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- [GitLab issue #243](#) System allows a user to skip entering values for a required metadata field while uploading a new document
- [GitLab issue #245](#) Add multiple metadata not possible

Mayan EDMS v2.0 release notes

Released: December 2015

Welcome to Mayan EDMS v2.0

What's new in Mayan EDMS v2.0

Update to Django 1.7

The biggest change of this release comes in the form of support for Django 1.7. Mayan EDMS makes use of several new features of Django 1.7 like: migrations, app config and transaction handling. The version of Django supported in this version is 1.7.10. With the move to Django 1.7, support for South migrations and Python 2.6 is removed. The switch to Django 1.7's app config means that the startup order of app should not longer have any relevance, cause any import or startup problems.

Frontend UI

The frontend UI HTML has been re-factored to use Bootstrap. Along with this update a lot of legacy HTML and CSS was removed, greatly simplifying the existing template and allowing the removal of some.

Theming and re-branding

All the presentation logic and markup has been moved into it's own app, the 'appearance' app. All modifications required to customize the entire look of the Mayan EDMS can now be done in a single app. Very little markup remains in the other apps, and it's usually because of necessity, namely the widgets.py modules.

Improved page navigation interface

Previously the document page interface used a fancybox windows leaving the current document in the background. This UI workflow as been improved and the document page navigation behaves like the rest of the document views.

Menu reorganization

To improve user experience, the main menu has been restructured based on function usage, moving seldom used buttons inside other views.

Removal of famfam icon set

The previously used icon set and icon display code was removed and a new system that favor font icon was added.

Document preview generation

The image conversion system was re-factored from the ground up and uses a much smarted caching system. The document image cache has it's own Django file storage driver and no longer default to the system /tmp directory. By moving the document image cache to a Django file storage, the cache doesn't need to reside in the same filesystem or even computer serving the document images. This change also allows nodes in a clustered install to share the document image cache.

Document submission for OCR changed to POST

Previously submitting a document for OCR could be done with a GET request to the corresponding URL. This design decision allowed for fast user experience but caused massive document submissions when sites were scanned by web spiders. The new workflow is to submit documents to the OCR queue only on POST request.

New YAML based settings system

The first phase of the new distributed settings system has landed in this version. This first change causes settings to be serialized to YAML. This also means that it is not possible to pass functions or custom classes as values to settings. Setting that related to a class or function, now specify the path to those classes or functions and they are imported dynamically at runtime. Example:

```
DOCUMENTS_STORAGE_BACKEND = 'storage.backends.filebasedstorage.FileBasedStorage'
```

Removal of auto admin creation

The auto admin user creation code used during new installs has been removed and it is its own reusable Django app. The app is available at <https://pypi.python.org/pypi/django-autodadmin>

Removal of dependencies

Through optimizations and code reduction several Python libraries and Django app are no longer required. These are:

- South
- GitPython
- django-pagination
- psutil
- python-hkp
- sendfile
- slate

ACL system re-factor

The Access Control System has been greatly simplified and optimized. The logistics to grant and revoke permissions are now as follows: Only Roles can hold permissions, groups and user can no longer on their own be granted a permission. Groups are now only organizational units that hold users and Roles are collections of groups. User are just a profile and authentication information object. So to grant a permission or access to a document to a user, grant those permissions to a new or existing role, add the desired user to a group and add that group to the role to which you granted the permission. When thinking about granting permissions think of it this way:

Permissions -> Roles -> Groups -> User

Permissions for a document -> Roles -> Groups -> User

Permissions for a type of document -> Roles -> Groups -> User

Object access control inheritance

A frequently asked feature is the ability to change the access control of a group of documents. This feature has been implemented in the form of object access control inheritance. This means that if you grant a permission to a role for a document type, that role will inherit that permission for all document that are later created of that type. If you revoke a permission from a role for a document type, that role loses that permission for all documents of that type. With this new system changing the access control of individual documents should be an edge case. This new ability of modifying the access control of document types is the new recommended method.

Removal of anonymous user support

Allowing anonymous users access to your document repository is no longer support. Administrators wanting to make a group of documents public are encouraged to create an user, group and role for that purpose.

Metadata validators re-factor

The metadata validators have been split into: Validators and Parsers. Validators will just check that the input value conforms to certain specification, raising a validation error is not and blocking the user from submitting data. The Parsers will transform user input and store the result as the metadata value.

Trash can support

To avoid accidental data loss, documents are not deleted but moved to a virtual trash can. From that trash can documents can them be deleted permanently. The deletion document documents and the moving of documents to the trash can are governed by two different permissions.

Retention policies

Support for retention policies was added and is control on a document type basis. Two aspects can be controlled: the time at which documents will be automatically moved to the trash can and the time after which documents in the trash can will be automatically deleted. By default all new document types created will have a retention policy that doesn't move documents to the trash can and that permanently deletes documents in the trash can after 30 days.

Support to share an index as a FUSE filesystem

Index mirror has been added after being removed several version ago. This time mirroring works by creating a FUSE filesystem that is then mounted anywhere in the filesystem. The previous implementation used symbolic links that while fast, required constant modification to keep in sync with the indexes structure and only worked when the document storage and the index mirror resided in the same physical computer or node. This new implementation allowing mirroring of indexes even across a network or if the document storage is not a traditional filesystem but a remote object store. Since this new FUSE mirroring uses direct read access to the database caching is provided and is controlled by the `MIRRORING_DOCUMENT_CACHE_LOOKUP_TIMEOUT` and `MIRRORING_NODE_CACHE_LOOKUP_TIMEOUT` setting options. Both setting have a default of 10 seconds.

Clickable preview images titles

To reduce the amount of clicks required to access a document, document previews titles are now clickable and will take the user straight to the document view.

Removal of eval

Use of Python's eval statement has been completely removed. Metadata type defaults, lookup fields, smart links and indexes templates now use Django's own template language.

Smarter OCR

Document OCR workflow has been improved to try to parse text for each document page and in failing to parse text will only perform OCR on that specific page, returning to the parsing behavior for the next page. This allowing proper text extraction of documents containing both, embedded text and images.

Failure tolerance

Previous versions made use of transactions to prevent data loss in the event of an unexpected error. This release improves on that approach by also reacting to infrastructure failures. Mayan EDMS can now recover without any or minimal data loss from critical events such as loss of connectivity to the database manager. This changes allow installation of using database managers that do not provide guaranteed concurrency such as SQLite, to scale to thousand of documents. While this configuration is still not recommended, Mayan EDMS will now work and scale much better in environments where parts of the infrastructure cannot be changed (such as the database manager).

For more information about this change read the blog post: <http://blog.robertorosario.com/testing-django-project-infrastructure-failure-tolerance/>

As a result of this work a new Django app called Django-sabot was created that gives Django projects the ability to create unit tests for infrastructure failure tolerance: <https://pypi.python.org/pypi/django-sabot>

RGB tags

Previously tags could only choose from a predetermined number of color. This release changes that and tags be of any color. Tags now store the color selected in HTML RGB format. Existing tags are automatically converted to this new scheme.

Default document type and default document source

After installation a default document type and document source are created, this means that users can start uploading documents as soon as Mayan EDMS is installed without having to do any configuration setting changes. The default document type and default document source are both called 'Default'.

Link unbinding

Support for allowing 3rd party apps to unbind links binded by the core apps was added to further improve re-branding and customization.

Statistics re-factor

Statistics gathering and generation has been overhauled to allow for the creation of scheduled statistics. This allows statistics computation to be scheduled during low load times. A new management command was added to purge stale or orphan schedules left behind by the editing of statistics scheduled. The command is *purgestatistics* and has no parameters.

Apps merge

Several app were merged to reduce complexity of the code based on function. These are: the *home*, *common*, *project_tools* and *project_setup* apps, as well as the *documents* and *document_acls* apps.

New signals

Two new signals are provided to better trigger processing documents at the correct moment, these are:

- common/perform_upgrade - Launched on the *performupgrade* management command to allow 3rd party apps to execute custom upgrade procedures in an unified manner.
- common/post_initial_setup - Launched on the *initialsetup* management command to allow for post install initialization or setup.
- common/post_upgrade - Launched after the *performupgrade* management command finishes.
- documents/post_version_upload = Launched after a new document version is uploaded.
- document/post_document_type_change = Launched after the document type of a document is changed.
- documents/post_document_created = Launched after a document is finally ready to be accessed, not when it is created.
- ocr/post_document_version_ocr - Launched when the OCR of a document version has finished.

Test improvements

Instead of a flat tests.py file, each app now has a tests/ directory containing tests modules for each particular aspect of an app, ie: test_models.py, test_views.py, test_classes.py. The total number and coverage of tests has been greatly increased.

Indexes recalculation

Indexes are now recalculated on when a new document is ready as well as when the metadata of a document changes. This allows indexing documents not only based on their metadata but also based on their properties.

Upgrade command

To reduce the steps and complexity of upgrades, the new *performupgrade* management command was added. All the upgrade steps will be performed by this command.

Admin changes

Installation admins are no longer required to have the *superusers* or *staff* Django account flags. All setup tasks are now governed by a permission which can be assigned to a role.

OCR functions split

The textual content of a document as interpreted by the OCR now resides as data in the *ocr* app and not in the *documents* app as before. OCR content might not be available for all documents after the upgrade and might need to be queued again. To help with this situation there is new tool called *OCR all documents* for this exact situation.

New internal document creation workflow

The new document upload code now returns a document stub while content is processing. This allows API users to have the document id of the document just uploaded and perform other actions on it while it becomes ready for access.

Auto logging

App logging to the console is now automatically enabled. If Django's DEBUG flag is *True* the default level for auto logging is *DEBUG*. If Django's DEBUG flag is *False* (as in production), the default level changes to *INFO*. This should make it easier to add relevant messages to issue tickets as well as a adequate logging during production.

Other changes

- Merge of document_print and document_hard_copy views.
- New class based and menu based navigation system.
- Re-purpose the installation app.
- New class based transformations.
- Usage of Font Awesome icons set.
- Move document text content display code to the OCR app.
- Add new permissions PERMISSION_OCR_CONTENT_VIEW.
- Document type OCR settings move to the OCR app.
- New dependencies:
 - PyYAML
 - django-autoadmin
 - django-pure-pagination
 - djangorestframework-recursive
- Management command to remove obsolete permissions: *purgepermissions*.
- Normalization of 'title' and 'name' fields to 'label'.
- Improved API, now at version 1.
- Invert page title/project name order in browser title.
- Use Django's class based views pagination.
- Reduction of text strings.
- OCR all documents.
- Add tool to OCR all documents of a type.

- Fix rendering of text files with Unicode characters.
- Capture body of emails as a text document.
- All app APIs are top level URLs.
- CI using gitlab-ci.
- Coverage report with codecov.io.
- Thumbnails for documents in trash.
- Production deployment documentation chapter.
- Command line to create an initial settings file: *createsettings*.
- Initialsetup now continues even if a settings/local.py exists.
- default_app_config for each app.
- Natural key support for many models allowing database migrations using dumped data.
- Separate documentation requirements file to allow for contributor who only want to work on documentation.
- Centralized testing with a new management command, *runtests*.
- Addition of a tox testing configuration.
- Email test body capture.
- Email subject and from values storage.
- Gitlab CI support.
- Codecov support.
- Improve text file rendering.
- Show other packages licenses.
- Task delay to allow DB replication.
- Automatic debug logging and info logging during production.

Removals

- Removal of the CombinedSource class.
- Removal of default class ACLs.
- Removal of the ImageMagick and GraphicsMagick converter backends.
- Remove support for applying roles to new users automatically.
- Removal of the DOCUMENT_RESTRICTIONS_OVERRIDE permission.
- Removed the page_label field.
- Removal of custom HTTP 505 error view.

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install -U mayan-edms
```

the requirements will also be updated automatically.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Migrate existing database schema with:

```
$ mayan-edms.py performupgrade
```

During the migration several messages of stale content types can occur:

```
The following content types are stale and need to be deleted:
```

```
XX | XX
```

Any objects related to these content types by a foreign key will also be deleted. Are you sure you want to delete these content types? If you're unsure, answer 'no'.

```
Type 'yes' to continue, or 'no' to cancel:
```

You can safely answer "yes" to all.

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

Remove unused dependencies:

```
$ pip uninstall South  
$ pip uninstall GitPython  
$ pip uninstall psutil  
$ pip uninstall python-hkp  
$ pip uninstall django-sendfile  
$ pip uninstall django-pagination  
$ pip uninstall slate
```

The upgrade procedure is now complete.

Backward incompatible changes

- Current document and document sources transformations will be lost during upgrade.
- Permissions and Access Controls granted to users and/or groups will be lost during upgrade.

Bugs fixed or issues closed

- GitHub issue #33 Update to Django 1.7
- GitHub issue #59 New bootstrap based UI
- GitHub issue #60 Backport class based navigation code from the unstable branch
- GitHub issue #62 Simplify and reduce code in templates
- GitHub issue #67 Python 3 compatibility: Update models `__unicode__` method to `__str__` methods (using Django's six library)
- GitHub issue #121 Twitter Bootstrap theme for Mayan EDMS
- GitHub issue #155 Header does not fit list on documents/list on small screens (laptop)
- GitHub issue #170 Remove use of python-hkp
- GitHub issue #182 Reorganize signal processors
- GitLab issue #131 error on initialsetup: GPG initialization error
- GitLab issue #135 Add document indexing filesystem mirroring
- GitLab issue #141 Merge common and main app
- GitLab issue #142 New authentication app
- GitLab issue #145 Convert document tags to user RGB value for code instead of predetermined choices
- GitLab issue #150 Add ‘trash can’ support
- GitLab issue #151 Add support for data retention policies
- GitLab issue #152 JSON API 500 error
- GitLab issue #154 /documents API endpoint should return document pk
- GitLab issue #155 Remove unused document page label field
- GitLab issue #156 Remove post OCR language cleanup
- GitLab issue #158 Django REST Swagger not working
- GitLab issue #159 Error during template rendering on /document/folder/add with non-admin user
- GitLab issue #160 Add audit logging
- GitLab issue #163 Removal of the compressed file support
- GitLab issue #164 Keep fancybox prev & next buttons enabled all the time
- GitLab issue #167 Add workflow completion number to states
- GitLab issue #168 Add field to store last error of source during execution
- GitLab issue #171 tesseract fails with german language (wrong abbreviation)
- GitLab issue #173 Add post_document_upload signal
- GitLab issue #174 Bootstrap UI with master branch

- [GitLab issue #176 Replace default email domain](#)
- [GitLab issue #177 Multi page tiff preview is not working](#)
- [GitLab issue #178 Add separate missing optional metadata and missing required metadata tools](#)
- [GitLab issue #181 Move task <-> queue assignment to apps.py](#)
- [GitLab issue #182 Document tags widget is not permissions aware](#)
- [GitLab issue #183 Separate metadata validators into: validators and parsers](#)
- [GitLab issue #184 Move literals in checkouts apps.py and tasks.py to literals.py](#)
- [GitLab issue #186 Scheduled task to delete all document stubs of more than X age.](#)
- [GitLab issue #187 Add tests for multi page tiff files](#)
- [GitLab issue #189 Use transient queues](#)
- [GitLab issue #190 Bump API version number](#)
- [GitLab issue #192 Use local model for document comments](#)
- [GitLab issue #197 Add continuous integration that is compatible with Gitlab](#)
- [GitLab issue #201 Untranslated items](#)
- [GitLab issue #202 AutoAdminSingleton matching query does not exist.](#)
- [GitLab issue #203 KeyError at /sources/upload/document/new/interactive/](#)
- [GitLab issue #204 Problems to add required metadata after changin the document type](#)
- [GitLab issue #216 Add default_app_config value to each app](#)
- [GitLab issue #223 \[Documents\] Trigger event_document_type_change on the model not on the view](#)
- [GitLab issue #227 decoder zip not available](#)
- [GitLab issue #228 Attribute error when trying to attach a tag for a user with inadequate permissions](#)
- [GitLab issue #229 Attribute error when a user tries to download a document - version 2.0.0b2](#)
- [GitLab issue #230 No option to create new document version even though user given permission in document ACL](#)
- [GitLab issue #231 User shown option to upload new version of a document even though it is blocked by checkout - v2.0.0b2](#)
- [GitLab issue #233 Available users instead of available groups](#)
- [GitLab issue #237 Forcefully checking in a document by a user without adequate permissions throws out an error](#)

1.0 series

Mayan EDMS v1.1.1 release notes

Released: May 21, 2015

What's new

Update to Django 1.6.11

The required version of Django has been updated to version 1.6.11.

Minor changes

- Fix make_dist.sh script
- Add test for issue #163
- Activate tests for the sources app
- Removal of the registration app
- New simplified official project description
- Improvements to the index admin interface
- Removal of installation statistics gathering
- Remove unused folder tag
- Fix usage of ugettext to ugettext_lazy
- Increase size of the lock name field
- New style documentation

Upgrading from a previous version

Using PIP

Type in the console:

```
$ pip install mayan-edms==1.1.1
```

the requirements will also be updated.

Using Git

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Common steps

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

None

Bugs fixed or issues closed

None

Mayan EDMS v1.1 release notes

Released: February 10, 2015

Welcome to Mayan EDMS v1.1

What's new in Mayan EDMS v1.1

Celery

All background tasks processing has been converted to use Celery. By default Mayan EDMS runs in “Eager” until a broker and result backend are configured as per [Celery’s documentation](#). This change made the built-in scheduler and job_processing apps obsolete, both were removed.

Views namespaces

All views are namespaced with the name of the app which defines them. If you have developed 3rd party apps for Mayan EDMS be sure to update any reference to a view by prepending the app name to the view name.

Removal of the splash screen

The static image home screen has been replaced with a quick links view, showing the most used actions: Uploading documents, viewing recent documents, viewing all documents and searching documents.

Sending and receiving documents via email

A link or entire documents can be sent as attachments via email. Documents can also be received via email with the addition of two document sources named IMAP and POP3 which correspond to the mail protocol used to fetch the documents. Read Django’s [email configuration settings documentation](#) for more details on how to set up mail serving.

Update to Django 1.6.8

Mayan EDMS has been updated to use Django 1.6.8.

Events app

The built-in history app has been removed in favor of a new events wrapper app for [Django](#) activity stream

Watch folders

Filesystem folders can be monitored for change and their files automatically uploaded as documents in Mayan EDMS.

Vagrant file included

A vagrant file is now included to allow developers to provision a virtual machine with the latest development version of Mayan EDMS.

User locale profile (language and timezone)

Interface language and locale setting can now be setup for each user and are not installation wide as before. Date and times offsets are automatically adjusted to each user's timezone settings.

Document states

A new simple workflow app that can represent document states has been included.

Explicit document types needed per index

Indexes can now be tied to document types, eliminating the need to update indexes for every document update. Indexes will only update when a document of the type to which they are associated is updated.

Optional and required metadata types

Metadata types can now be assigned in two ways to documents types, as optional or required. Values for required metadata types as the name implies, must be entered for documents to be able to be uploaded. Optional metadata types on the other hand can be left blank by the user.

Bulk document type change

It is now possible to change the document type of previously uploaded documents. When the document type of a document is changed the metadata values are reset and the metadata types of the new document type are automatically assigned.

New release cycle

Starting with this version a new release cycle methodology will come into effect. The goal of this release cycle is to allow two series of versions of Mayan EDMS to be active at a given time: A new major version with new functionality and a minor version providing upgrades and fixes. This release (1.1) will be active and supported during releases of versions 2.x, but will go into end-of-life as soon as version 3.0 is released, at which time version series 2.x will go into maintenance mode.

Deprecation of Python 2.6

Series 1.0 of Mayan EDMS will be the last series supporting Python 2.6. Series 2.0 will be using Django 1.7.x which itself requires Python 2.7 or later.

Improved testings

Mayan EDMS is now automatically tested against SQLite, MySQL and PostgreSQL.

API updates

Many new API endpoints have been added exposing the majority of Mayan EDMS functionality.

Messages contextual improvements

Many updates and simplifications were made to the source text messages to reduce the difficulty of translating Mayan EDMS and maintaining the contextual meaning of the text messages.

Improved method for custom settings

Custom settings now use a string based value, it is longer needed to import classes when customizing a setting:

```
from custom_app.backends import CustomStorageBackend  
DOCUMENTS_STORAGE_BACKEND = CustomStorageBackend
```

Instead the fully qualified name of the class must be passed as the setting value:

```
DOCUMENTS_STORAGE_BACKEND = 'custom_app.backends.CustomStorageBackend'
```

Removal of the OCR config setting

OCR behavior is now a document type property meaning that it can be turned on or off for specific document types.

Per document language setting

Previously the document language used for OCR was specified for the entire installation. If documents in multiple languages were uploaded some suffered lower success rates. Now the language of each document can be specified.

Metadata validation and parsing support

It is now possible to create functions to validate metadata value input or parse and store corrected values. Three sample metadata validations functions are included: Parse date and time, Parse date and Parse time.

Removal of 960 grid system in favor Pure CSS's grid system

By using Pure CSS's columns based grid system, the move towards a Bootstrap UI migration has advanced greatly.

Simplified UI

All user actions as well as the logout button are now under the user functions section.

Stable PDF previews generation

The way PDF were being generated has been improved greatly eliminating spurious segmentation faults at the expense of a small speed penalty.

More technical documentation

Many new sub topics were added to the development section of the documentation to allow developers to better understand the inner workings and philosophies of Mayan EDMS.

Other changes

- Removal of the MAIN_SIDE_BAR_SEARCH setting
- Removal of THEMES and themes support
- Removal of VERBOSE_LOGIN setting
- Removal of graphics backend supported file format list view
- Removal of the MAIN_DISABLE_ICONS setting
- Removal of specialized Sentry support
- Removal of the MAIN_ENABLE_SCROLL_JS setting
- Remove hardcoded root (/) redirections
- Removal of APScheduler as a requirement
- Removal of the scheduled jobs view
- Removal of the web_theme app
- Removal of the sources icon selection support
- Removal of the in-app help panels
- Removal of the duplicate document search feature
- Removal of filesystem document indexes mirroring feature
- Improve sources app model sub classes and inheritance handling

- Addition of CORS support to the API

Upgrading from a previous version

IMPORTANT! Before running the upgrade make sure none of your documents have duplicated metadata types, meaning that the same metadata type must not appear twice for any given document.

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Migrate existing database schema with:

```
$ mayan-edms.py migrate dynamic_search 0001 --fake  
$ mayan-edms.py migrate history 0001 --fake  
$ mayan-edms.py migrate linking 0001 --fake  
$ mayan-edms.py migrate lock_manager 0001 --fake  
$ mayan-edms.py migrate tags 0001 --fake  
$ mayan-edms.py migrate
```

During the migration several messages of stale content types can occur:

The following content types are stale and need to be deleted:

```
metadata | documenttypedefaults  
metadata | metadataset  
metadata | metadatasetitem  
ocr | documentqueue  
ocr | queuedocument  
sources | watchfolder  
sources | outofprocess  
sources | webform  
sources | stagingfolder  
tags | tagproperties
```

Any objects related to these content types by a foreign key will also be deleted. Are you sure you want to delete these content types? If you're unsure, answer 'no'.

Type 'yes' to continue, or 'no' to cancel:

You can safely answer "yes".

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

Remove unused dependencies:

```
$ pip uninstall APScheduler  
$ pip uninstall django-taggit
```

The upgrade procedure is now complete.

Backward incompatible changes

- You will have to redefine your document sources due to the new extended models for this app.
- Check your configurations of smart links and indexes to use the newly provided arguments.

Bugs fixed or issues closed

- issue #30 Document workflows
- issue #32 Watched folders
- issue #34 Postprocessing document queue
- issue #35 Metadata validation
- issue #37 Add from `__future__ import unicode_literals`
- issue #39 Capitalization of messages
- issue #40 Update references to root ('/') path
- issue #46 Advanced search past 1st page
- issue #49 Problems with large pdf files
- issue #50 raise CommandNotFound(path)
- issue #51 Search with ANONYMOUS error
- issue #55 Document approval cicle?
- issue #56 Removal of non essential features, views, models
- issue #57 Migrate to Celery for task query and periodic tasks
- issue #64 Pluralize messages properly
- issue #65 Backport the ability to receive documents via email
- issue #66 Python 3 compatibility: Add from `__future__ import unicode_literals` and remove all u'`
- issue #68 Revise and update the use gettext vs. gettext_lazy
- issue #69 Feature removal: remove "Unregistered" message from the title bar
- issue #71 Add retry support to the converter task
- issue #72 Delete unused static icons
- issue #74 Cache a document's first document version
- issue #75 Move automatic OCR queueing from a configuration settings to a property of Document Type model
- issue #77 Add document view permission support to the search app
- issue #78 COMMON_TEMPORARY_DIRECTORY seems not to be used everywhere
- issue #79 Error installing
- issue #82 Make document type a required field
- issue #83 Simplify source app views and navigation

- issue #84 Remove template context variable ‘object_name’ to improve translations
- issue #85 Reset page count for a single document
- issue #86 Move migrations to new ‘south_migrations’ folders
- issue #87 Per document language selection
- issue #88 Remove metadata type selection from the upload wizard
- issue #89 Allow metadata types to be required for specific document types
- issue #90 Remove the app_registry app
- issue #91 Don’t preserve the ?page= URL query string value when switching sources during document upload
- issue #92 Make register_multi_item_links class aware
- issue #95 Installation error on Mac OSX; OSError: [Errno 2] No such file or directory
- issue #96 Remove hard code User model references
- issue #97 Make multi item links a drop down list
- issue #104 Finish polishing metadata validation patch
- issue #105 Tie smart links setups to document types
- issue #106 Convert document indexing app actions to Celery
- issue #107 Restrict document metadata addition and removal
- issue #108 New home screen
- issue #109 Add Roles API endpoints
- issue #111 Add Checkouts API endpoints
- issue #112 Add OCR API endpoints
- issue #114 Implement UI language as user preference
- issue #116 Add documentation topic explicitly noting the binary requirements
- issue #118 When a metadata type is removed from a document type, remove it from all the documents of that type
- issue #119 When a required metadata type is added to a document type, add it to all documents of that type
- issue #126 Failing migration with SQLite
- issue #127 Failing migration with Postgres
- issue #128 Add Indexes API endpoints
- issue #129 Search api shouldn’t memorize requested page as part of the query
- issue #130 Users API is not working correctly
- issue #131 Is there an API to update a user’s password?
- issue #137 Enhancement of language selection
- issue #138 Possibility to keep zoom factor
- issue #139 Translatability of language selection
- issue #140 Thumbnail creation for ods crashing
- issue #143 Exception Value: ‘exceptions.ValueError’ object has no attribute ‘messages’

- issue #144 Behavior of ‘Edit metadata’ (Recent Documents)
- issue #146 Periodic task not initiated for mail boxes and watch folders
- issue #149 Attribute error in document download
- issue #150 Double second menu entry
- issue #152 Document content empty
- issue #153 south migration with postgres: documents: 031_remove_orphan_documents
- issue #154 south migration with postgres: tags: 0002_auto__add_tag__chg_field_tagproperties_tag
- issue #157 upload new version of a document not working
- issue #158 Plural form not matching singular form in ocr app bug i18n

Mayan EDMS v1.0 release notes

August 26, 2014

Welcome to Mayan EDMS v1.0!

Overview

After a year of low activity the focus of this release was to get the code updated to work with the most recent version of Django and the most recent version of the dependencies Mayan EDMS uses. The goal was to have a strong and stable 1.0 release so reduction, trimming, warning reductions and bug fixes were the primary focus of this cycle. Because of this not much emphasis was placed on adding new features, or at least new features that could have the potential to break things.

What's new in Mayan EDMS v1.0

New home

The growth and reach of the project has necessitated for a while the move of the project to its own organization in Github. This move has finally been done, the new URL of the project is: <https://github.com/mayan-edms/mayan-edms>

Django 1.6

This release updates the required version of Django to 1.6, bringing with it not only new features, but also a lot of security updates, a new project directory structure and new deployment methods.

Translation updates

The translation for all languages were synchronized to the latest transifex project sources. Translation completion as reported by Transifex:

- English - 100%
- Spanish - 100%
- Arabic - 96%

- Bosnian - 96%
- French - 96%
- German - 96%
- Romanian - 96%
- Russian - 96%
- Italian - 77%
- Portuguese - 76%
- Dutch - 68%
- Portugese (Brazil) - 67%
- Bulgarian - 62%
- Danish - 42%
- Vietnamese - 40%
- Polish - 39%
- Hungarian - 27%
- Indonesian - 18%
- Slovenian - 17%
- Persian - 6%
- Croatian - 3%
- Turkish - 3%

Model updates

There were some convenience properties created to allow quick access to a document's version and pages. These custom properties were removed and an official method to access these properties as provided by Django is now used.

Source code improvements

A circular import of metadata and document_index apps code from the documents app was removed. Document index updates are now handled via signals, not called directly as before. Hundreds of PEP8 style fixes, unused import removals, unused variables removals and removal of remarked or unused code. Removal of the DEVELOPMENT flag (was used to trigger static media serving during development), this is now handled by the DEBUG flag. The DEBUG flag is now set to *True* by default as per Django 1.6 defaults. Removed usage of Django's JSON libraries using Python's JSON library instead. Update of time and date use to use Django's new timezone aware data and time handling. Removal of custom code in favor of using modules provided by Django or by existing 3rd party libraries. Unification of code used for equal or similar purpose in various modules.

Removal of included 3rd party modules

One last 3rd party module was included with the source code of Mayan EDMS. This module is now available on PyPI and fetched during the installation instead of being included.

Automatic testing and code coverage check

Some initial tests were added, which will help with the detection of regressions or bugs when adding new features. More tests are needed, but the initial work has been started.

Update of required modules and libraries versions

Many of the required modules and libraries have been updated to a more recent version if not to their most recent released version.

Database connection leaks

Stale database connection being left open by scheduler tasks are now explicitly closed. This avoids consumption of the pool of database connections, increases stability and reduces memory usage.

Deletion of detached signatures

Detached signatures can now be deleted, if accidentally added to the wrong document.

Removal of fabric based installations

These files are now part of their own project and located at <https://github.com/mayan-edms/mayan-fabric>

Pluggable OCR backends

A commonly requested feature, it is now possible to write backends drivers to do document OCR using software or services other than Tesseract.

OCR improvements

OCR queue state is now reset when reloading Mayan EDMS, avoiding the OCR queue to remain locked. *unpaper* binary is now an optional pre OCR requirement, the OCR queue will now continue working if *unpaper* is not installed. Addition of post OCR processing support for French and German.

License change

Mayan EDMS is now licensed under the Apache 2.0 license. This means many things but the main change is that inclusion of Mayan EDMS into commercial products is now explicitly allowed.

PyPI package

Mayan EDMS has been packaged and submitted to the [PyPI](#) Python Package Index making it even easier to install and use.

New REST API

This release feature a completely new REST API and automatic API documentation. This new API is also used internally by *Mayan EDMS* itself.

Other changes

More office document types are now recognized and supported. More file types are now supported as text files and properly previewed and parsed. Removal of the legacy runserver.sh and runserver_plus.sh scripts. New document preview generation and display pipeline, faster, simpler. Inclusion of a proof of concept compressed storage backend.

Upgrading from a previous version

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next upgrade/add the new requirements:

```
$ pip install --upgrade -r requirements.txt
```

Migrate existing database schema with:

```
$ mayan-edms.py migrate acls 0001 --fake  
$ mayan-edms.py migrate taggit 0001 --fake  
$ mayan-edms.py migrate ocr 0001 --fake  
$ mayan-edms.py migrate
```

Add new static media:

```
$ mayan-edms.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- If using the SQLite3 database option, rename the file named *mayan.sqlite* file to *db.sqlite3* and move it to the new *media* provided folder.
- Also move to the *media* folder any *gpg_home*, *document_storage* and *image_cache* folders from your previous installation.

Bugs fixed or issues closed

- None

0.13 release

Mayan EDMS v0.13 release notes

December 2012

Welcome to Mayan EDMS v0.13!

Overview

Initially this version was meant as a the third maintenance release of the 0.12 series, but with the amount of changes and updates that were included it was obvious this was not just a bug fix version anymore hence the version jump to 0.13 instead of 0.12.3

What's new in Mayan EDMS v0.13

Django 1.3.5

This release updates the required version of Django to 1.3.5 to take advantage of the security fixes added to that version of the framework.

Bulk document tagging and untagging

As requested by issue [issue #31](#) this feature has been implemented and enabled in this version. Attaching or removing tags from a large number of documents is now much easier.

Registration

Based on requests made by the community for greater commercial support and services for Mayan EDMS, a new feature has been added that allows users to register their copies of Mayan EDMS and better help users with commercial support packages. Registration for non commercial users is voluntary and optional, and in no way affects the functionality of Mayan EDMS. However even for non commercial users registration offers the advantage of automatically branding the user's copy of Mayan EDMS with their name or the company name in the title area.

Per document type indexing

Index can now be restricted to update only on specific document types, this greatly increases the usefulness of indexes, and prevents unwanted index updates.

Bootstrap

Setting up Mayan EDMS after installation has been identified by users as the main difficulty when knowledge about Mayan EDMS is relatively low. To address this situation a new feature that provides preconfigured setups has been added. These preconfigured setups are published in the Mayan EDMS website and upon synchronization are available to users, this gives users access and integrators access to new setups without having to wait for new versions of Mayan EDMS. Aside from including preconfigured setups, the new bootstrap app has the ability of dumping an user's current setup into a serialized text file which can be tweaked by hand and sent via email to other users. The possibilities enabled by this range from company wide defaults setups to consultants providing their clients with

customized setups without having to access their clients' Mayan EDMS instances. JSON, YAML and a custom YAML format (<http://djangosnippets.org/snippets/2461/>) are supported by this new app.

Add documents to folders in bulk

As requested, the ability to add more than one document at a time to a selected folder has been added.

Translation updates

The translation for all the current languages were synchronized to the latest transifex project sources.

Model updates

Several small fixes to the behavior of some model were added, namely that the names of some models should be unique. The document type name as well as the metadata set name were two models that were updated to behave this way.

Navigation updates

There were some additions and changes to the navigation to make it more intuitive or to add an alternate way to access the same information better. The bulk attachment of tags was one of these. Previously users were added or removed from groups, now users can be assigned to groups without leaving the user view. The name of an existing metadata set can now be edited and what was previously called metadata set edit is now more aptly named metadata members which adds or removes metadata types into a single organizational unit. An error that caused a duplicate menu link in the document type filename view was also fixed.

Support for UNOCONV dropped

Support for converting office documents by calling LibreOffice via UNOCONV has been disabled for a while ever since LibreOffice start including document conversion support from the command line. This version completely removes any traces of code and configuration options related to UNOCONV.

Optimizations

Inspired by the idea of getting Mayan EDMS running effectively on low power hardware such as the Raspberry Pi, several rounds of profiling and optimization were done.

Some caching optimization were introduced to the permission model, resulting in a speed increase of 33% in rendering time on views with heavy permission checking and a 18% query reduction on cache hits.

Upgrading from a previous version

If you installed Mayan EDMS by cloning the Git repository issue the commands:

```
$ git reset --hard HEAD  
$ git pull
```

otherwise download the compressed archived and uncompress it overriding the existing installation.

Next add the new requirements:

```
$ pip install -r requirements/production.txt
```

Migrate existing database schema with:

```
$ ./manage.py migrate document_indexing
$ ./manage.py migrate metadata 0001 --fake
$ ./manage.py migrate metadata
$ ./manage.py migrate bootstrap
$ ./manage.py migrate documents
$ ./manage.py migrate registration
```

Add new static media:

```
$ ./manage.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed or issues closed

- issue #28 “Document indexing based on filesystem problem”
- issue #29 Default ACL documentation
- issue #30 “Ubuntu 12.0.4 LTS complaining about mising ‘fabfile/templates/settings_local.py’ (master branch)”
- issue #31 “Bulk add tags Feature request”
- issue #32 “Error run \$python manage.py syncdb”
- issue #33 “Fetches translations from Transifex. “
- issue #34 “Initial syncdb fails because of import time database access”
- issue #35 “Fix first install syncdb on Postgres”
- issue #37 “When 19 >= &document_type_id >= 10 in url, error page not found”
- issue #39 Feature request “Bulk move folders”

0.12 release

Mayan EDMS v0.12.2 release notes

June 2012

This is the second maintenance release of the 0.12 series.

Overview

As with the previous release bug fixes and minor feature were the focus for this release too. [issue #24](#) has been fixed and document check outs have been added too as per the feature request posted as [issue #26](#). The way the history events for a document are presented has been improved and it is now more useful as it provides filtering by event type. To improve the diagnosis of installation of runtime error a simple view showing the number of internal interval jobs being used by Mayan EDMS as well as a new app which shows a detail of the current installation environment were added.

What's new in Mayan EDMS v0.12.2

Smarter auto admin creation

Mayan EDMS creates a administrator user during the database creation phase to reduce the amount of steps required for a functional install. The creation of this account is controlled by the configuration option *COMMON_AUTO_CREATE_ADMIN*, the username of the account is specified with the configuration option *COMMON_AUTO_ADMIN_USERNAME* and the password of this account by the option *COMMON_AUTO_ADMIN_PASSWORD*. Previously the *COMMON_AUTO_ADMIN_PASSWORD* defaulted to ‘admin’ which created an administrator account of username ‘admin’ with a password of ‘admin’. The new default is to randomize an initial password and show this password at the login screen until the administrator password is changed.

Document check outs

As per the feature request filed under [issue #26](#), a new document check out and check in functionality has been added. Users can now check out a document and lock new version of it from being uploaded to avoid editing conflicts. Document check outs have an expiration period after which Mayan EDMS will automatically check them back in to avoid a permanent document lockout. Only the user who has checked out a document can upload new versions of it or check the document back in before the expiration period, unless being granted the `Allow overriding check out restrictions` or `Forcefully check in documents` permission respectively.

Installation environment app

Diagnosing remote installations of web based applications without access to the command line can be a bit hard, to alleviate this situation a new installation environment details app has been added. The purpose of this app is to provide support staff information about the physical environment where Mayan EDMS has been installed. To avoid possible security compromises only administrators can access this app.

Editable compressed documents filename

Previously when downloading more than one document in a compressed manner, Mayan EDMS would produce a file with the name `document_bundle.zip` for download. A different filename can now be specified at the same download dialog.

German translation

A German language translation has been added thanks to Tetja Rediske and Tilmann Sittig.

Statistics gathering

Previous attempts at gathering usage statistics have been met with deficient results. User participation in surveys as well as the quality of the data entered by users was disappointing. That is why this version of Mayan EDMS features an anonymous statistics gathering functionality.

Upgrading from a previous version

Start off by adding the new requirements:

```
$ pip install -r requirements/production.txt
```

Migrate existing database schema with:

```
$ ./manage.py migrate common 0001 --fake
$ ./manage.py migrate common
$ ./manage.py migrate checkouts
$ ./manage.py migrate installation
```

Add new static media:

```
$ ./manage.py collectstatic --noinput
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed

- issue #24 “Duplicated filename extension when uploading a new version of a document”
- issue #26 “checkout feature request”

Stuff removed

- Feedback app

Mayan EDMS v0.12.1 release notes

May 2012

This is the first maintenance release of the 0.12 series.

Overview

While bug fixes and minor feature were the focus for this release, some bigger changes were included because of their importance. The parsing of documents saw a complete rewrite being now class based and allows for more than one parser per mimetype with sequential fallback. This provides the best text extraction on deployments where users have

control over the installation and basic extraction when deploying on the cloud or other environments where users don't have the ability to install OS level binaries.

What's new in Mayan EDMS v0.12.1

Fabric file (fabfile)

A Fabric file is included to help users not very familiar with Ubuntu, Python and Django install Mayan EDMS, or for system administrators looking to automate the install whether in local or remote systems. At the moment the fabfile will install Mayan EDMS in the same configurations listed in this documentation, that is: (Ubuntu/Debian/Fedora) + virtualenv + Apache + MySQL. Feel free to submit your configuration settings and files for different databases, webserver or Linux distribution. More configurations will be added to the fabfile as more are tested.

Documentation update

The installation instructions were updated to include the installation of the libpng-dev and libjpeg-dev libraries as well as the installation of the poppler-utils package. An additional step to help users test their new installation of Mayan EDMS was also added.

Translations

The Italian translation has been synchronized with the source files at Transifex and finished to %100 completion.

Usability improvements

The index instance view now feature the same multi document action buttons (Submit to OCR, delete, download, etc) as the mail and recent document views.

Better office document conversion

A new method of converting office documents has been implemented, this new method doesn't require the use of the command line utility UNOCONV. If this new method proves to work better than previous solutions the use of UNOCONV may be deprecated in the future. The conversion method adds just one new configuration option: *CONVERTER_LIBREOFFICE_PATH* which defaults to '/usr/bin/libreoffice'.

Better PDF text parsing

Brian E. submitted a patch to use the Poppler package pdftotext utility to extract text from PDF files. This is now the default method Mayan EDMS will execute to try to extract text from a PDF and failing that will fallback to the previous method. This change add a new configuration option: *OCR_PDFTOTEXT_PATH* to specify the location of the pdftotext executable, it defaults to '/usr/bin/pdftotext'. Be sure to install the `poppler-utils` os package to take advantage of this new parser.

Changed defaults

The OCR queue is now active by default when first created during the syncdb phase and the *OCR_AUTOMATIC_OCR* option now defaults to True. These two changes are made to reduce the steps required for new users to start enjoying the benefits of automatic text extraction from uploaded documents without having to read the documentation and have a more functional default install.

Upgrading from a previous version

Start off by adding the new requirements:

```
$ pip install -r requirements/production.txt
```

Migrate existing database schema with:

```
$ ./manage.py migrate documents
```

Install the poppler-utils package:

- Ubuntu, Debian:

```
$ apt-get install -y poppler-utils
```

- Fedora:

```
$ yum install -y poppler-utils
```

The upgrade procedure is now complete.

Backward incompatible changes

- None

Bugs fixed

- Issue #25 “Office document conversion error”

Stuff removed

- None

Mayan EDMS v0.12 release notes

February 2012

Welcome to Mayan EDMS v0.12!

This release commemorates Mayan EDMS first anniversary!

Overview

Aside from new features, the focus of this release of Mayan EDMS also been about improving the code and documentation quality standard even further. The permission system has been completely overhauled to make it entire class based. The other big change is the addition of object level permissions, with this new system being applied to documents, folder, tags and smart links. There is also a small batch of navigation improvements. Big code cleanup and lots of changes ‘under the hood’, most of these are not visible to the end user, but make the code cleaner and more manageable so that more and better features can be added in future releases:

- Absolute imports used throughout the code
- All app permissions have been moved to a separate permissions.py file per app
- Complete permission system refactor.
- Document signing code move to its own app
- Initial unit tests
- A lot of logging used throughout the entire project.
- Much functionality moved to model managers.
- A lot of code converted into classes.
- Coding style improvements.
- Template user authentication state logic improvements, for stronger prevention against intrusion or unintentional display or access of restricted data.
- Removal of remarked code.

What's new in Mayan EDMS v0.12

ACL support

- Object level access control is now in place for documents, folders, tags and smart links. What this means is that administrators can now grant permissions to users, groups or roles on specific objects. A more in-depth explanation of how this new ACL system can be found in the [3 tier access control](#) section of the permissions chapter.
- Default class ACL support. Administrators can setup the access control lists that new documents, folders and tags will automatically inherit when created. Aside from assigning permission to users, groups and roles to specific objects, there is a special user called *Creator*, used to allow the access control list that the actual creator of an object will inherit.

Anonymous user support

Anonymous user support is a two tier function, first is the addition of the `COMMON_ALLOW_ANONYMOUS_ACCESS` configuration option that allows non authenticated user to browse all the pages of a Mayan EDMS installation. The second part of this support is the ability to assign permissions or individual access to objects to anonymous users.

Translations

A new Italian translation is available, provided by SeeOpen.IT (www.seeopen.it, info@seeopen.it) as well as complete Russian translation update by Sergei Glita. Included in this release also the initial translation to Polish by mic.

Usability improvements

- Detached signature behavior improved, uploading a new detached signature erases the previous one.
- Usability improvement in the role member's add/removal form, by using HTML's option groups tag property

2 Step download process

The code for downloading single and multiple document and document versions has been merged with compression support also added. This allows for the download of documents in their original format or compressed and well as the download of several documents in a single compressed file.

Customizable GPG home directory

Addition of the *SIGNATURES_GPG_HOME* configuration option to let administrators set Mayan EDMS's GPG instance home directory, used to store keyrings and other GPG configuration files.

Out of process bulk uploading

A management command has been added to help upload a large number of documents from a compressed file. For information about this new feature check the Initial data loading chapter.

Out of process user import

A management command has been added to import a large number users from a CSV file. More information about this new feature can also be found in the Initial data loading chapter.

Refactored document indexing

The document indexing functionality has been improved and moved from experimental stage to beta stage. Index configuration menus are now available on the Setup menu and allows administrators to create skeleton trees that will be populated with document links depending on their metadata and properties. These populated trees can also be mirrored on the physical filesystem and shared using Samba or another filesharing server giving users a structured view of the documents contained within Mayan EDMS from the Indexes tab or from a mirrored index shared via the network. A new configuration option has been added, *DOCUMENT_INDEXING_FILESYSTEM_SERVING*, which maps the index internal name with the physical directory where such index will be mirrored on disk.

Help shape Mayan EDMS's future

Included in this version is a small feedback application, found under the About main menu, where users by just answering some questions can help determine the priority of the next planned features on the pipeline, or even help add new features if enough requests are received. All questions are optional but answering as many as possible will help greatly understand the need of the Mayan EDMS user base.

Staging file previews

The staging file previews now show the filename for easier identification and speedier upload selection. The staging files previews are now treated as a gallery which means that users can preview an entire page of staging files without having to click and close each one individually.

Upgrading from a previous version

Start off by adding the new requirements:

```
$ pip install -r requirements/production.txt
```

Then create the new database structures with:

```
$ ./manage.py syncdb
```

Afterwards migrate existing database schema with:

```
$ ./manage.py migrate permissions 0001 --fake  
$ ./manage.py migrate permissions
```

When the following message appears

```
The following content types are stale and need to be deleted:
```

```
permissions | permission
```

Any objects related to these content types by a foreign key will also be deleted. Are you sure you want to delete these content types? If you're unsure, answer 'no'.

```
Type 'yes' to continue, or 'no' to cancel:
```

Type yes and press **Enter**

And continue migrating database schema with:

```
$ ./manage.py migrate documents  
$ ./manage.py migrate document_signatures  
$ ./manage.py migrate folders 0001 --fake  
$ ./manage.py migrate folders  
$ ./manage.py migrate document_indexing 0001 --fake  
$ ./manage.py migrate document_indexing  
$ ./manage.py migrate sources 0001 --fake  
$ ./manage.py migrate sources
```

Again when a similar message appears

```
The following content types are stale and need to be deleted:
```

```
document_indexing | indexinstance
```

Any objects related to these content types by a foreign key will also be deleted. Are you sure you want to delete these content types? If you're unsure, answer 'no'.

```
Type 'yes' to continue, or 'no' to cancel:
```

Type yes and press **Enter**

The upgrade procedure is now complete.

Backward incompatible changes

The permission system has been completely reworked so sadly this is a place where even data migration can't help and the permissions assigned to roles will be lost during the upgrade to version 0.12. Users, groups and roles will be preserved only permissions need to be assigned again, so write down your role permission setup before upgrading.

Bugs fixed

- Issue #17, special thanks to Dave Herring for all the help including access to a machine suffering with the issue, and to Sergei Glita for his research and eventual find of the core cause.
- Statistics fixes.
- Fixed get_image_cache_name regression in the OCR app.

Stuff removed

- Support for Celery and Sentry has been drop for now.
- Removed the ‘db_index’ argument from Text fields definition and migrations as it was causing error messages for MySQL users, thanks to Sergei Glita for reporting this one.
- Configuration options removed:
 - OCR_CACHE_URI
 - DOCUMENT_INDEXING_FILESYSTEM_FILESERVING_PATH - Use the newest *DOCUMENT_INDEXING_FILESYSTEM_SERVING*
 - DOCUMENT_INDEXING_FILESYSTEM_FILESERVING_ENABLE - Use the newest *DOCUMENT_INDEXING_FILESYSTEM_SERVING*

0.11 release

Version 0.11.1

- Fixed a document deletion regression
- Improves error detection when importing keys from a keyserver, catching the exception KeyImportError and not KeyFetchingError
- Fixes a wrong method call when verifying signatures for the first time upon document uploading
- django-compress is now disabled by default to avoid problems when deploying with DjangoZoom
- Improve post metadata set delete redirection

Version 0.11

- Support for signed documents verification added, embedded and detached signatures are supported. When verifying a document Mayan EDMS will try to fetch the public key from the list of keyservers provided in the configuration option SIGNATURES_KEYSERVERS (which defaults to ‘pool.sks-keyservers.net’). A public key management view has been added to the setup menu as well as a key query and fetching view to manually import keys from a keyserver.
- Added support for document versions. Users can upload unlimited amount of versions for a document using a very flexible document version numbering system, users can also revert to a previous document version.
- OCR queue processing improvements.
- Office documents handling improvements.
- Text extraction support for office documents.
- RTF text documents are now handled as office documents.
- Added a view to delete the document image cache, useful when switching converter backends or doing diagnostics.
- Added South to the requirements.
- Merged documents’ filename and extension database fields into a single filename field, filename are stored as uploaded not manipulation is done. Users with existing data must install South and run the appropriate migrate commands:

```
$ pip install -r requirements/production.txt
$ ./manager syncdb
$ ./manage.py migrate documents 0001 --fake
$ ./manage.py migrate documents
```

- Added new office document mimetype * application/vnd.ms-office
- Fixed documents not saving the file encoding
- Removed extra slash in ajax-loader.gif URL fixes #15, thanks to IHLeanne for finding this one

0.10 release

Version 0.10.1

- Upgraded django-compressor to version 1.1.1, run:

```
$ pip install --upgrade -r requirements/production.txt
```

to upgrade

- django-compressor is now disabled by default, users must explicitly enable it adding COMPRESS_ENABLED=True to their settings_local.py file

Version 0.10

- Added a proper setup views for the document grouping functionality.
- Document grouping is now called smart linking as it relates better to how it actually works. The data base schema was changed and users must do the required:

```
$ ./manager syncdb
```

for the new tables to be created.

- Grappelli is no longer required as can be uninstalled.
- New smarter document preview widget that doesn't allow zooming or viewing unknown or invalid documents.
- New office document converter, requires:
 - LibreOffice (<https://www.libreoffice.org/>)
 - unoconv [version 0.5] (<https://github.com/dagwieers/unoconv>)
- The new office documents converter won't convert files with the extension .docx because these files are recognized as zip files instead. This is an issue of the libmagic library.
- New configuration option added CONVERTER_UNOCONV_USE_PIPE that controls how unoconv handles the communication with LibreOffice. The default of True causes unoconv to use **pipes**, this approach is slower than using **TCP/IP** ports but it is more stable.
- Initial *REST API* that exposes documents properties and one method, this new *API* is used by the new smart document widget and requires the package `djangorestframework`, users must issue a:

```
$ pip install -r requirements/production.txt
```

to install this new requirement.

- MIME type detection and caching performance updates.
- Updated the included version of `jQuery` to 1.7
- Updated the included version of `JqueryAsyncImageLoader` to 0.9.7
- Document image serving response now specifies a MIME type for increased browser compatibility.
- Small change in the scheduler that increases stability.
- Russian translation updates Sergey Glita
- Improved and generalized the OCR queue locking mechanism, this should eliminate any possibility of race conditions between Mayan EDMS OCR nodes.
- Added support for signals to the OCR queue, this results in instant OCR processing upon submittal of a document to the OCR queue, this works in addition to the current polling processing which eliminates the possibility of stale documents in the OCR queue.
- Added multiple document OCR submit link
- Re enabled tesseract language specific OCR processing and added a one (1) time language neutral retry for failed language specific OCR

0.9 release

Version 0.9.1

- Added handling percent encoded unicode query strings in search URL, thanks to Sergei Glita for reporting.
- Added a FAQ explaining how to fix MySQL collation related error when doing searches also thanks to Sergei Glita for reporting this one.

Version 0.9

- Simplified getting mimetypes from files by merging 2 implementations (document based and file based)
- Updated python converter backend, document model and staging module to use the new get_mimetype API
- Only allow clickable thumbnails for document and staging files with a valid image
- Removed tag count from the group document list widget to conserve vertical space
- Updated required Django version to 1.3.1
- Removed the included 3rd party module django-sendfile, now added to the requirement files.
 - User should do a pip install -r requirements/production.txt to update
- Changed to Semantic Versioning (<http://semver.org/>), with recommendations 7, 8 and 9 causing the most effect in the versioning number.
- Added Russian locale post OCR cleanup backend Sergei Glita
- Reduced severity of the messages displayed when no OCR cleanup backend is found for a language
- Complete Portuguese translation (Emerson Soares and Renata Oliveira)
- Complete Russian translation Sergei Glita)
- Added animate.css to use CSS to animate flash messages with better fallback on non JS browsers
- The admin and sentry links are no longer hard-coded (Meurig Freeman)
- Improved appearance of the document tag widget (<https://p.twimg.com/Ac0Q0b-CAAE1lfA.png>:large)
- Added django_compress and cssmin to the requirements files and enabled django_compress for CSS and JS files
- Added granting and revoking permission methods to the permission model
- Correctly calculate the mimetype icons paths when on development mode
- Added a new more comprehensive method of passing multiple variables per item in multi item selection views
- Used new multi parameter passing method to improve the usability of the grant/revoke permission view, thanks to Cezar Jenkins (<https://twitter.com/#!/emperorcezar>) for the suggestion
- Added step to the documentation explaining how to install Mayan EDMS on Webfaction
- Added an entry in the documentation to the screencast explaining how to install Mayan EDMS on DjangoZoom
- Added required changes to add Mayan EDMS to Transifex.com
- Fixed the apache contrib file static file directory name
- Added improved documentation

0.8 release

Version 0.8.3

- Added a Contributors file under the docs directory
- Moved the document grouping subtemplate windows into a document information tab
- Change the mode the setup options are shown, opting to use a more of a dashboard style now
- Changed the tool menu to use the same button layout of the setup menu

- Moved OCR related handling to the tools main menu
- Improved the metadata type and metadata set selection widget during the document upload wizard
- Added a view to the about menu to read the LICENSE file included with Mayan
- Added converter backend agnostic image file format descriptions
- Disable whitelist and blacklist temporarily, removed document_type field from interactive sources
- Fully disabled watch folders until they are working correctly
- Updated the project title to ‘Mayan EDMS’
- If ghostscript is installed add PDF and PS to the list of file formats by the python converter backend
- Use Pillow (<http://pypi.python.org/pypi/Pillow>) instead of PIL
 - Pillow is a fork of PIL with several updated including better jpeg and png library detection
 - Users must uninstall PIL before installing Pillow
- Updated the static media url in the login exempt url list
- Added remediatory code to sidestep issue #10 caused by DjangoZoom’s deployment script executing the collectstatic command before creating the database structure with syncdb. Thanks to Joost Cassee (<https://github.com/jcassee>) for reporting this one.
- Perform extra validation of the image cache directory and fallback to creating a temporary directory on validation failure
- Fixed a source creation bug, that caused invalid links to a non existing source transformation to appear on the sidebar

Version 0.8.2

- Moved code to Django 1.3
 - Users have to use the collectstatic management command:

```
$ ./manage.py collectstatic
```
 - The site_media directory is no more, users must update the media serving directives in current deployments and point them to the static directory instead
- The changelog is now available under the about main menu
- Grappelli no longer bundled with Mayan
 - Users must install Grappelli or execute:

```
$ pip install --upgrade -r requirements/production.txt
```
- Even easier UI language switching
- Added email login method, to enable it, set:

```
AUTHENTICATION_BACKENDS = ('common.auth.email_auth_backend.EmailAuthBackend',)  
COMMON_LOGIN_METHOD = 'email'
```

Version 0.8.1

- Tags can now also be created from the main menu
- Added item count column to index instance list view
- Updated document indexing widget to show different icon for indexes or indexes that contain documents
- Replaced the Textarea widget with the TextAreaDiv widget on document and document page detail views
 - This change will allow highlighting search terms in the future
- Unknown document file format page count now defaults to 1
 - When uploading documents which the selected converted backend doesn't understand, the total page count will fallback to 1 page to at least show some data, and a comment will be automatically added to the document description
- Added new MAIN_DISABLE_ICONS to turn off all icons
 - This option works very well when using the default theme
- The default theme is now activo
- Improved document page views and document page transformation views navigation
- Added OCR queue document transformations
 - Use this for doing resizing or rotation fixes to improve OCR results
- Added reset view link to the document page view to reset the zoom level and page rotation
- Staging files now show a thumbnail preview instead of preview link

Version 0.8

- Distributed OCR queue processing via celery is disabled for the time being
- Added support for local scheduling of jobs
 - This addition removes celery beat requirement, and make is optional
- Improve link highlighting
- Navigation improvements
- Documents with an unknown file format now display a mime type place holder icon instead of an error icon
- Mayan now does pre caching of document visual representation improving overall thumbnail, preview and display speed
 - Page image rotation and zooming is faster too with this update
- Removed all QUALITY related settings
- COMMON_TEMPORARY_DIRECTORY is now validated when Mayan starts and if not valid falls back to creating its own temporary folder
- Added PDF file support to the python converter backend via ghostscript
 - This requires the installation of:
 - * ghostscript python package
 - * ghostscript system binaries and libraries
- Added PDF text parsing support to the python converter backend

- This requires the installation of:
 - * pdfminer python package
- Added PDF page count support to the python converter backend
- Added python only converter backend supporting resizing, zooming and rotation
 - This backend required the installation of the python image library (PIL)
 - This backend is useful when Graphicsmagick or Imagemagick can not be installed for some reason
 - If understand fewer file format than the other 2 backends
- Added default transformation support to document sources
- Removed DOCUMENT_DEFAULT_TRANSFORMATIONS setup options
- Document sources are now defined via a series of view under the setup main menu
- This removes all the DOCUMENT_STAGING related setup options
 - Two document source types are supported local (via a web form), and staging
 - However multiple document sources can be defined each with their own set of transformations and default metadata selection
- Use python-magic to determine a document's mimetype otherwise fallback to use python's mimetypes library
- Remove the included sources for python-magic instead it is now fetched from github by pip
- Removed the document subtemplates and changed to a tabbed style
- Added link to document index content view to navigate the tree upwards
- Added new option MAIN_DISABLE_HOME_VIEW to disable the home main menu tab and save some space
- Added new option to the web theme app, WEB_THEME_VERBOSE_LOGIN that display a more information on the login screen (version, copyright, logos)
- Added a confirmation dialog to the document tag removal view

0.7 release

Version 0.7.6

- Added recent searches per user support
 - The amount of searches stored is controlled by the setup option SEARCH_RECENT_COUNT
- The document page zoom button are now disabled when reaching the minimum or maximum zoom level
- The document page navigation links are now disabled when view the first and last page of a document
- Document page title now displays the current page vs the total page count
- Document page title now displays the current zoom level and rotation degrees
- Added means set the expansion compressed files during document creation, via web interface removing the need for the configuration options: UNCOMPRESS_COMPRESSED_LOCAL_FILES and UNCOMPRESS_COMPRESSED_STAGING_FILES
- Added 'search again' button to the advances search results view
- Implementes an advanced search feature, which allows for individual field terms

- Search fields supported: document type, MIME type, filename, extension, metadata values, content, description, tags, comments

Version 0.7.5

- Added a help messages to the sidebar of some views
- Renamed some forms submit button to more intuitive one
 - ‘Search’ on the submit button of the search form
 - ‘Next step’ on the document creation wizard
- Added view to list supported file formats and reported by the converter backend
- Added redirection support to multi object action views
- Renamed ‘document list’ link to ‘all documents’ and ‘recent document list’ to ‘recent documents’
- Removed ‘change password’ link next to the current user’s name and added a few views to handle the current user’s password, details and details editing

Version 0.7.4

- Renamed ‘secondary actions’ to ‘secondary menu’
- Added document type setup views to the setup menu
- Added document type file name editing views to the setup menu
- Fixed document queue properties sidebar template not showing

Version 0.7.3

- Refactored main menu navigation and converted all apps to this new system
- Multi item links are now displayed on top of generic lists as well as on the bottom
- Spanish translation updates
- Updated requirements to use the latest development version of django-mptt
- Improved user folder document removal views
- Added ability to specify default metadata or metadataset per document type
- Converted filename handling to use os.path library for improved portability
- Added edit source object attribute difference detection and logging to history app
- Missing metadata type in a document during a multi document editing doesn’t raise errors anymore.
 - This allows for multi document heterogeneous metadata editing in a single step.
- Added document multi item links in search results
 - Direct editing can be done from the search result list
- Permissions are now grouped and assigned a group name
- Improved role management views
- Document type is now an optional document property

- Documents can be created without an explicit document type
- Added support for per user staging directories
- Updated logos

Version 0.7

- Added confirmation dialogs icons
- Added comment app with support for adding and deleting comments to and from documents
- Updated requirements files as per issue #9
- Show tagged item count in the tag list view
- Show tagget document link in the tags subtemplate of documents
- Made comment sorted by oldest first, made comment subtemplate scrollable
- Rename comments app to document_comment to avoid conflict with Django's comment app
- Made document comments searchable

0.5 release

Version 0.5.1

- Applied initial merge of the new subtemplate renderer
- Fixed tag removal logic
- Initial commit to support document comments
- Updated so that loading spinner is displayed always
- Exclude tags from the local document upload form
- Added document tagging support
 - Requires installing `django-taggit` and doing a `sync-db`

Version 0.5

- Added tag list view and global tag delete support
- Added tag editing view and listing documents with a specific tag
- Changed the previewing and deleting staging files views to required DOCUMENT_CREATE permission
- Added no-parent-history class to document page links so that iframe clicking doesn't affect the parent window history
 - Fixes back button issue on Chrome 9 & 10
- Added per app version display tag
- Added loading spinner animation
- Messages tweaks and translation updates
- Converter app cleanups, document pre-cache, magic number removal

- Added OCR view displaying all active OCR tasks from all cluster nodes
- Disabled `CELERY_DISABLE_RATE_LIMITS` by default
- Implement local task locking using Django locmem cache backend
- Added doc extension to office document format list
- Removed redundant transformation calculation
- Make sure OCR in processing documents cannot be deleted
- PEP8, pylint cleanups and removal of relative imports
- Removed the obsolete `DOCUMENTS_GROUP_MAX_RESULTS` setting option
- Improved visual appearance of messages by displaying them outside the main form
- Added link to close all notifications with one click
- Made the queue processing interval configurable by means of a new setting: `OCR_QUEUE_PROCESSING_INTERVAL`
- Added detection and reset of orphaned ocr documents being left as ‘processing’ when celery dies
- Improved unknown format detection in the graphicsmagick backend
- Improved document conversion API
- Added initial support for converting office documents (only ods and docx tested)
- Added sample configuration files for supervisor and apache under contrib/
- Avoid duplicates in recent document list
- Added the configuration option `CONVERTER_GM_SETTINGS` to pass GraphicsMagicks specific commands to the GM backend
- Lower image conversion quality if the format is jpg
- Inverted the rotation button, more intuitive this way
- Merged and reduced the document page zoom and rotation views
- Increased permissions app permission’s label field size
 - DB Update required
- Added support for metadata group actions
- Reduced the document pages widget size
- Display the metadata group numeric total in the metadata group form title
- Reorganized page detail icons
- Added first & last page navigation links to document page view
- Added interactive zoom support to document page detail view
- Spanish translation updates
- Added `DOCUMENTS_ZOOM_PERCENT_STEP`, `DOCUMENTS_ZOOM_MAX_LEVEL`, `DOCUMENTS_ZOOM_MIN_LEVEL` configuration options to allow detailed zoom control
- Added interactive document page view rotation support
- Changed the side bar document grouping with carousel style document grouping form widget

- Removed the obsolete DOCUMENTS_TRANSFORMATION_PREVIEW_SIZE and DOCUMENTS_GROUP_SHOW_THUMBNAIL setting options
- Improved double submit prevention
- Added a direct rename field to the local update and staging upload forms
- Separated document page detail view into document text and document image views
- Added grab-scroll to document page view
- Disabled submit buttons and any buttons when during a form submit
- Updated the page preview widget to display a infinite-style horizontal carousel of page previews
- Added support user document folders
 - Must do a syncdb to add the new tables
- Added support for listing the most recent accessed documents per user
- Added document page navigation
- Fixed diagnostics url resolution
- Added confirmation dialog to document's find missing document file diagnostic
- Added a document page edit view
- Added support for the command line program pdftotext from the poppler-utils packages to extract text from PDF documents without doing OCR
- Fixed document description editing
- Replaced page break text with page number when displaying document content
- Implemented detail form readonly fields the correct way, this fixes copy & paste issues with Firefox
- New document page view
- Added view to add or remove user to a specific role
- Updated the jQuery packages with the web_theme app to version 1.5.2
- Made AVAILABLE_INDEXING_FUNCTION setting a setting of the documents app instead of the filesystem_serving app
- Fixed document download in FireFox for documents containing spaces in the filename
- If mime detection fails set mime type to “ instead of ‘unknown’
- Use document MIME type when downloading otherwise use ‘application/octet-stream’ if none
- Changed the way document page count is parsed from the graphics backend, fixing issue #7
- Optimized document metadata query and display
- Implemented OCR output cleanups for English and Spanish
- Redirect user to the website entry point if already logged and lands in the login template
- Changed from using SimpleUploadedFile class to stream file to the simpler File class wrapper
- Updated staging files previews to use sendfile instead of serve_file
- Moved staging file preview creation logic from documents.views to staging.py
- When deleting staging file, it's cached preview is also deleted
- Added a new setup option:

- FILESYSTEM_INDEXING_AVAILABLE_FUNCTIONS - a dictionary to allow users to add custom functions
- Made automatic OCR a function of the OCR app and not of Documents app (via signals)
 - Renamed setup option DOCUMENT_AUTOMATIC_OCR to OCR_AUTOMATIC_OCR
- Clear node name when requeueing a document for OCR
- Added support for editing the metadata of multiple documents at the same time
- Added Graphics magick support by means of user selectable graphic conversion backends
 - Some settings renamed to support this change:
 - * CONVERTER_CONVERT_PATH is now CONVERTER_IM_CONVERT_PATH
 - * CONVERTER_IDENTIFY_PATH is now CONVERTER_IM_IDENTIFY_PATH
 - Added options:
 - * CONVERTER_GM_PATH - File path to graphicsmagick's program.
 - * CONVERTER_GRAPHICS_BACKEND - Backend to use: ImageMagick or GraphicMagick
- Raise ImportError and notify user when specifying a non existant converter graphics backend
- Fixed issue #4, avoid circular import in permissions/__init__.py
- Add a user to a default role only when the user is created
- Added total page count to statistics view
- Added support to disable the default scrolling JS code included in web_theme app, saving some KBs in transfer
- Clear last ocr results when requeueing a document
- Removed the 'exists' column in document list view, diagnostics superceded this
- Added 3rd party sendfile app (support apache's X-sendfile)
- Updated the get_document_image view to use the new sendfile app
- Fixed the issue of the strip spaces middleware conflicting with downloads
- Removed custom IE9 tags
- Closed Issue #6
- Allow deletion of non existing documents from OCR queue
- Allow OCR requeue of pending documents
- Invalid page numbers now raise Http404, not found instead of error
- Added an additional check to lower the chance of OCR race conditions between nodes
- Introduce a random delay to each node to further reduce the chance of a race condition, until row locking can be implemented or is implemented by Django
- Moved navigation code to its own app
- Reimplemented OCR delay code, only delay new document Added a new field: delay, update your database schema accordingly
- Made the concurrent ocr code more granular, per node, every node can handle different amounts of concurrent ocr tasks Added a new field: node_name, update your database schema accordinging
- Reduced default ocr delay time

- Added a new diagnostics tab under the tools menu
- Added a new option `OCR_REPLICATION_DELAY` to allow the storage some time for replication before attempting to do OCR to a document
- Added OCR multi document re-queue and delete support
- Added simple statistics page (total used storage, total docs, etc)
- Implemented form based and button based multi item actions (button based by default)
- Added multi document delete
- Fixed a few HTML validation errors
- Issues are now tracked using github
- Added indexing flags to ocr model
- Small optimization in document list view
- Small search optimization
- Display “DEBUG mode” string in title if DEBUG variable is set to True
- Added the fix-permissions bash script under misc/ folder
- Plugged another file descriptor leak
- Show class name in config settings view
- Added missing config option from the setup menu
- Close file descriptor to avoid leaks
- Don’t allow duplicate documents in queues
- Don’t raise `PermissionDenied` exception in `PermissionDenied` middleware, even while debugging
- Fixed page number detection
- Created ‘simple document’ for non technical users with all of a document pages content
- Use document preview code for staging file also
- Error picture literal name removal
- Spanish translation updates
- Show document file path in regards of its storage
- Added new setting: side bar search box
- Implemented new `PermissionDenied` exception middleware handler
- Permissions app api now returns a `PermissionDenied` exception instead of a custom one
- Added new 403 error template
- Updated the 404 template to display only a not found message
- Moved the login required middleware to the common app
- Fixed search app’s `model.objects.filter` indentation, improved result count calculation
- Added dynamic comparison types to search app
- Separated search code from view code
- Correctly calculate show result count for multi model searches

- Fixed OCR queue list showing wrong thumbnail
- Fixed staging file preview
- Show current metadata in document upload view sidebar
- Show sentry login for admin users
- Do not reinitialize document queue and/or queued document on reentry
- Try extra hard not to assign same uuid to two documents
- Added new transformation preview size setting
- Renamed document queue state links
- Changed ocr status display sidebar from form based to text based
- Added document action to clear all the document's page transformations
- Allow search across related fields
- Optimized search for speed and memory footprint
- Added LIMIT setting to search
- Show search elapsed time on result page
- Converter now differentiates between unknown file format and convert errors
- Close file descriptors when executing external programs to prevent/reduce file descriptor leaks
- Improved exception handling of external programs
- Show document thumbnail in document ocr queue list
- Make ocr document date submitted column non breakable
- Fix permissions, directories set to mode 755 and files to mode 644
- Try to fix issue #2, “random ORM field error on search while doing OCR”
- Added configurable location setting for file based storage
- Prepend storage name to differentiate config options
- Fixed duplicated document search
- Optimized document duplicate search
- Added locale middleware, menu bar language switching works now
- Only show language selection list if localemiddleware is active
- Spanish translation updates
- Added links, views and permissions to disable or enable an OCR queue
- Enabled Django’s template caching
- Added document queue property side bar window to the document queue list view
- Added HTML spaceless middleware to remove whitespace in HTML code
- If current user is superuser or staff show thumbnail & preview generation error messages
- Added a setting to show document thumbnail in metadata group list
- Started adding configurations setting descriptions
- Initial GridFS storage support

- Implemented size and delete methods for GridFS
- Implement GridFS storage user settings
- Added document link in the OCR document queue list
- Link to manually re queue failed OCR
- Don't separate links (enclose object list links with white-space: nowrap;)
- Added document description to the field search list
- Sort OCR queued documents according to submitted date & time
- Document filesystem serving is now a separate app
 - Steps to update (Some warnings may be returned, but these are not fatal as they might be related to missing metadata in some documents):
 - * rename the following settings:
 - DOCUMENTS_FILESYSTEM_FILESERVING_ENABLE to FILESYSTEM_FILESERVING_ENABLE
 - DOCUMENTS_FILESYSTEM_FILESERVING_PATH to FILESYSTEM_FILESERVING_PATH
 - DOCUMENTS_FILESYSTEM_SLUGIFY_PATHS to FILESYSTEM_SLUGIFY_PATHS
 - DOCUMENTS_FILESYSTEM_MAX_RENAME_COUNT to FILESYSTEM_MAX_RENAME_COUNT
 - * Do a ./manage.py syncdb
 - * Execute ‘Recreate index links’ locate in the tools menu
 - * Wait a few minutes
- Added per document duplicate search and a tools menu option to seach all duplicated documents
- Added document tool that deletes and re-creates all documents filesystem links
- Increased document’s and document metadata index filename field’s size to 255 characters
- Added sentry to monitor and store error for later debugging
- Zip files can now be uncompressed in memory and their content uploaded individually in one step
- Added support for concurrent, queued OCR processing using celery
- Apply default transformations to document before OCR
- Added unpaper to the OCR conversion pipe
- Added views to create, edit and grant/revoke permissions to roles
- Added multipage documents support (only tested on pdfs)
 - To update a previous database do: [d.update_page_count() for d in Document.objects.all()]
- Added support for document page transformation (no GUI yet)
- Added permissions and roles support
- Added python-magic for smarter MIME type detection (<https://github.com/ahupp/python-magic>).
- Added a new Document model field: file_mime_encoding.
- Show only document metadata in document list view.
- If one document type exists, the create document wizard skips the first step.
- Changed to a liquid css grid

- Added the ability to group documents by their metadata
- New abstracted options to adjust document conversion quality (default, low, high)

CHAPTER 5

Concepts

Introductions to all the key parts of Mayan EDMS you'll need to know:

Document types

The basic unit of data in Mayan EDMS is the `document_type`. A document type can be interpreted also as a document category, a document class, or a document template. Document types need to be created before documents can be uploaded. It is not possible to upload documents without assigning them a document type. Examples of document type: **invoices, blueprints, receipts**.

Settings and attributes are applied to document types and documents will inherit those settings and attributes based on the document type they were assigned when uploaded into Mayan EDMS. A document can only be of one type at a given moment, but if needed, the type of a document can be changed. Upon changing its type, the document will lose its previous settings and attributes, and will inherit the settings and attributes of its new type.

Metadata

Metadata is the name of the attribute of a document. The concept of metadata is divided in two: **metadata types** (size, color, distance) and **metadata values** for those types. Metadata types are defined in the setup menu and associated with document types. Then when a document is uploaded, a value for that metadata can be entered. There are two kinds of metadata type to document type relations: optional and required. When a metadata type is optional for a document type, it can be left blank for a document being uploaded and the upload will still be successful. On the other hand required metadata type must be given a value or it will not be possible to upload the document at hand.

Examples of metadata type: Invoice number, color, employee id.

The data entry of metadata types can be set to allow any value to be provided (the default) or a list of possible values can be entered in the `Lookup` configuration option and users will be presented with a drop down list of options instead of the default text entry box.

If metadata types are setup to allow any value to be entered a validation option can be chosen to block the entry of invalid data. Metadata types also provide parsers which will not block the entry of data but are able to interpret

and modify the value provided by the user to conform to a specific format. An example of a provided parser is the date parser which will interpret and correct dates provided by users regardless of the format in which they are entered.

Permissions

Mayan EDMS provides very fine control over which actions users can perform. Action control works by allowing roles, that are composed of groups of users to be granted a permission such that the holder of that permission can exercise it throughout the entire system.

In other words, users themselves can't hold a permission, permissions are granted only to roles. Users can't directly belong to a role, they can only belong to a group. Groups can be members of roles. Roles are system permission units and groups are business organizational units.

Sources

Document sources define places from which documents can be uploaded or gathered.

The current document sources supported are:

- Web - HTML forms with a `Browse` button that will open the file dialog when clicked to allow selection of files in the user's computer to be uploaded as documents.
- POP3 email - Provide the email, server and credential of a POP3 based email to be scanned periodically for email. The body of the email is uploaded as a document and the attachments of the email are uploaded as separate documents.
- IMAP email - Same as the POP3 email source but for email accounts using the IMAP protocol.
- Watch folder - A filesystem folder that is scanned periodically for files. Any file in the watch folder is automatically uploaded.
- Staging folder - Folder where networked attached scanners can save image files. The files in these staging folders are scanned and a preview is generated to help the process of upload. Staging folders and Watch folders work in a similar way with the main difference being that Staging folders are interactive while Watch folders are automatic; documents in a Watch folder are uploaded periodically and documents in a Staging folder remain indefinitely there until an user uploads them. A preview for files in a Staging folder is also provided. An example of Staging folder use is when multiple people are scanning documents but only one person must be allowed to upload those documents. This one person examines the scans quality and decides what to upload and what to reject and have re-scanned. Watch folders can be used when the quality of the scans is irrelevant or when they will be known to be of good quality, such as when receiving e-faxes as PDFs.

Document source can be configured to allow document bundles to be uploaded as compressed files which are decompressed and their content uploaded as separate documents. This feature is useful when migrating from another document manager system.

Access control lists

Besides the permissions system explained in [Permissions](#), Mayan EDMS provides per object permission granting. This feature is used to grant a permission to a role, but this permission can only be executed for a limited number of objects (documents, folders, tags) instead of being effective system-wide.

Example:

In this scenario only users in groups belonging to the Accountants role would be able to view the 2015 Payroll report.txt document.

Inherited access control

It is also possible to grant a permission to a role for a specific document type (*Document types*). Under this scheme all users in groups belonging to that role will inherit that permission for all documents of that type.

Example:

The role Accountants is given the permission document view for the document type Payroll reports. Now all users in groups belonging to the Accountants role can view all documents of the type Payroll reports without needing to have that permissions granted for each particular Payroll reports type document.

If access control for the Payroll reports documents needs to be updated it only needs to be done for the document type and not for each document of the type Payroll reports.

Transformations

Transformation are persistent manipulations to the previews of the stored documents. For example: a scanning equipment may only produce landscape PDFs. In this case an useful transformation for that document source would be to rotate all documents scanned by 270 degrees after being uploaded, this way whenever a document is uploaded from that scanner it will appear in portrait orientation. In this case add a this transformation to the Mayan EDMS source that is connected to that device this way all pages scanned via that source with inherit the transformation as they are created.

Transformations can also be added to existing documents, by clicking on a document's page, then clicking on "transformations". In this view the Actions menu will have a new option that reads "Create new transformation". At the moment the rotation, zoom, crop, and resize transformations are available. Once the document image has been corrected resubmit it for OCR for improved results.

Transformations are not destructive and do not physically modify the document file, they just modify the document's graphical representation.

Checkouts

Checkouts are a way to block certain accesses or actions of a document for a period of time.

An user can choose to checkout a document to work on an update and block new versions of that document to be uploaded by other users. Document are checked out for a certain amount of time and if not manually checked in by the original user, will be checked in automatically by the system.

To be able to check in documents that were checked out by other users, the permission 'Forcefully check in documents' is required.

Document versioning

Mayan EDMS has the ability to store different versions of the same document. A comment field is provided to allow users to summarize the new version changes in comparison with the previous one. If a new version was uploaded by mistake or such new version is no longer necessary the option to revert to a previous version of the document is provided.

Only the interactive document sources (*Sources*) (Web and Staging folders) are available to upload new document versions.

Document signatures

Mayan EDMS supports two types of document signatures: embedded and detached signatures. When a document with an embedded signature is uploaded, this signature is readily detected as part of the document inspection step. The status of the signature can be verified by accessing the signatures sections of a document.

Existing non signed documents can be signed in one of two ways: by downloading the document, signing it, and uploading the signed document as a new version of the existing one or by creating a detached signature for the non signed document and uploading such detached signature file.

Maintenance of the public keyring can be done using the Key management functionality in the Setup menu.

From this menu, key servers can be queried and the results imported. Public keys no longer needed can also be deleted from this menu.

Only GNU Privacy Guard signatures are support at the moment.

Only version 1 of GNU Privacy Guard is supported for now.

OCR backend

Mayan EDMS ships an OCR backend that uses the FLOSS engine Tesseract (<https://github.com/tesseract-ocr/tesseract/>), but it can use other engines. To support other engines crate a wrapper that subclases the OCRBackendBase class defined in mayan/apps/ocr/classes. This subclass should expose the execute method. For an example of how the Tesseract backend is implemented take a look at the file mayan/apps/ocr/backends/tesseract.py

Once you create you own backend, in your local.py settings add the option OCR_BACKEND and point it to your new OCR backend class path.

The default value of OCR_BACKEND is "ocr.backends.tesseract.Tesseract"

To add support to OCR more languages when using Tesseract, install the corresponding language file. If using a Debian based OS, this command will display the available language files:

```
apt-cache search tesseract-ocr
```

Indexes

Indexes are an automatic method to hierarchically organize documents in relation to their properties (*Metadata*, label, MIME type, etc). To use indexes you need to first create an index template. Once created, associate the index to one or more *Document types*.

Index are hierarchical models so a tree template needs to be specified for them. This tree template will contain references to document metadata or properties that will be replaced with the actual value for those metadata or properties.

Example:

- Document type: Product sheet
- Metadata type: Product year, associated as a required metadata for the document type Product sheet.
- Index: Product sheets per year, and associated to the document type Product sheet.

- Index slug: `product-sheets-per-year`. Slugs are internal unique identifiers that can be used by other Mayan EDMS modules to reference each index.
- Index tree template as follows:

Now every time a new `Product sheet` is uploaded a hierarchical unit with the value of the metadata type `Product year` is created and a link to the uploaded `Product sheet` added to it.

Example:

Suppose three `Product sheets` are uploaded with the following values as their `Product year` metadata: 2001, 2002, 2001 respectively. The result index that will be generated based on the tree template would be as follows:

Mirroring

Indexes can be exported as FUSE filesystems. Using the management command `mountindex` we could export the previous example index as follows:

```
mkdir -p ~/indexes/products
mayan-edms.py mountindex product-sheets-per-year ~/indexes/products
```

The `~/indexes/products` directory will now have a directory and files structure identical to that of the index. Once indexes are mounted with this command, they behave like any other filesystem directory and can even be further shared via the network with network file system software like [Samba](#) or [NFS](#).

Indexes and mirrored indexes are Read Only as they are generated as a result of prior activities like document uploads, metadata changes.

Languages

The list of languages choices in the language dropdown used for documents is based on the current ISO 639 list. This list can be quite extensive. To reduce the number of languages available use the settings `DOCUMENTS_LANGUAGE_CHOICES`, and set it to a nested list of abbreviations + languages names like:

```
DOCUMENTS_LANGUAGE_CHOICES = (('eng', 'English'), ('spa', 'Spanish'))
```

The default language to appear on the dropdown can also be configured using:

```
DOCUMENTS_LANGUAGE = 'spa'
```

Use the correct ISO 639-3 language abbreviation (https://en.wikipedia.org/wiki/ISO_639) as this code is used in several subsystems in Mayan EDMS such as the OCR app to determine how to interpret the document.

Smart links

Smart links are a way to link documents without changing how they are organized in their respective indexes. Smart links are useful when two documents are related somehow but are of different type or different hierarchical units.

Example: A patient record can be related to a prescription drug information document, but they each belong to their own [Indexes](#).

Smart links are rule based, but don't create any organizational structure. Smart links just show the documents that match the rules as evaluated against the metadata or properties of the currently displayed document.

Indexes are automatic hierarchical units used to group documents, smart links are automatic references between documents.

Example:

- Document type: Patient records
- Metadata type: Prescription, associated as an optional metadata for the document type Patient records.
- Document type: Prescription information sheets

A smart link with the following condition, will automatically links patient records to the prescription information sheets based on the value of the metadata type of the patient record.

```
foreign label is equal to {{ document.metadata_value_of.prescription }}
```

Tags

Tags allow giving documents a binary property. Documents can also be tagged with more than one tag. Once tagged, documents can be searched also by their tags and from the tags main menu a list of all the documents with a particular tag can be obtained easily. Aside from their texts, tags can be assigned a particular color.

Mailing documents

To be able to send documents via email from inside Mayan EDMS you need to add and configure the following configuration variables in your mayan/settings/local.py file:

```
EMAIL_HOST = 'smtp.gmail.com' # Or similar
EMAIL_PORT = 587
EMAIL_HOST_USER = '<your smtp username>'
EMAIL_HOST_PASSWORD = '<your smtp password>'
EMAIL_USE_TLS = True
```

“Mail is sent using the SMTP host and port specified in the EMAIL_HOST and EMAIL_PORT settings. The EMAIL_HOST_USER and EMAIL_HOST_PASSWORD settings, if set, are used to authenticate to the SMTP server, and the EMAIL_USE_TLS and EMAIL_USE_SSL settings control whether a secure connection is used.”

For more details consult Django’s documentation on the topic: <https://docs.djangoproject.com/en/1.8/ref/settings/#email-host>

Settings

When Mayan EDMS is initially installed a local.py file is created inside the /mayan/settings/ folder. So if you installed Mayan EDMS according to the instructions provided in this documentation your local.py should be located in the directory: /usr/share/mayan-edms/mayan/settings/local.py.

For a list of all the configuration options, go to “Setup” then “Settings” on your browser. This is also a good place to check if your overrided setting option value in your local.py file is being interpreted correctly.

File storage

The files are stored and placed under Mayan EDMS “control” to avoid filename clashes each file gets renamed to its UUID (Universally Unique ID), without extension, and stored in a simple flat arrangement in a directory.

This doesn’t stop access to the files but renaming, moving or updating directly them is not recommended because it would throw the database out of sync.

Because Mayan EDMS components are as decoupled from each other as possible, storage in this case is decoupled and its behavior is controlled not by the project but by the `Storage` module class. All the other modules don’t make any assumptions about how the actual document files are stored. This way files can be saved locally, over the network or even across the Internet and everything will still operate exactly the same.

The default file storage backend: `storage.backends.filebasedstorage.FileBasedStorage` is a simple backend that only supports paths and not IP addresses. In case you are interested in using remote volumes to store documents (NFS, SAMBA), first mount these volumes so that they appear as a directories to Mayan EDMS. For direct support for remote volumes a custom backend would be needed such as those provided by the Django Storages project (<https://django-storages.readthedocs.org/en/latest/>).

Backups

To backup your install of Mayan EDMS just copy the actual document files and the database content. If you are using the default storage backend, the document files should be found in `mayan/media/document_storage/`.

To dump the content of your database manager refer to the documentation chapter regarding database data “dumping”.

Example:

- Postgresl: <http://www.postgresql.org/docs/current/static/backup.html>
- MySQL: <https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html>
- SQLite: Just copy the file `mayan/media/db.sqlite3`

CHAPTER 6

Development

Mayan EDMS is under active development, and contributions are welcome.

If you have a feature request, suggestion or bug report, please open a new issue on the [GitLab issue tracker](#). To submit patches, please send a merge request on [GitLab](#).

Project philosophies

How to think about Mayan EDMS when doing changes or adding new features, why things are the way they are in Mayan EDMS.

- Functionality must be as market/sector independent as possible, code for the 95% of use cases.
- Each user must be able to configure and customize it to their needs after install.
- Abstract as much as possible, each app must be an expert in just one thing, for other things they should use the API/classes/functions of other apps.
- Assume as little as possible about anything outside the project (hardware, OS, storage).
- Provide Python based abstraction so that a default install runs with a single step.
- No hard dependencies on binaries unless there is no other choice.
- Provide “drivers” or switchable backends to allow users to fine tune the installation.
- Call to binaries only when there is no other choice or the Python choices are not viable/mature/efficient.
- Each app is as independent and self contained as possible. Exceptions, the basic requirements: navigation, permissions, common, main.
- If an app is meant to be used by more than one other app it should be as generic as possible in regard to the project and another app will bridge the functionality.
 - Example: since indexing (document_indexing) only applies to documents, the app is specialized and depends on the documents app.

Coding conventions

Follow PEP8

Whenever possible, but don't obsess over things like line length.

```
$ flake8 --ignore=E501,E128,E122 | less
```

Imports

Import order should be:

- Standard Python modules
- Installed Python modules
- Core Django modules
- Installed Django modules
- Mayan EDMS modules
- Local imports

Example:

```
from __future__ import absolute_import

# Standard Python library
import base64

# 3rd party installed Python libraries
import requests

# Django core modules
from django.db.models import Q
from django.template.defaultfilters import slugify
from django.utils.translation import ugettext
from django.utils.translation import ugettext_lazy as _

# 3rd party installed Django libraries
from rest_framework import APIView

# Mayan apps
from metadata.classes import MetadataClass

# Local app imports (relative)
from .conf.settings import (
    AVAILABLE_INDEXING_FUNCTIONS,
    MAX_SUFFIX_COUNT, SLUGIFY_PATHS
)
from .exceptions import MaxSuffixCountReached
from .filesystem import (
    fs_create_index_directory, fs_create_document_link,
    fs_delete_document_link, fs_delete_index_directory,
    assemble_suffixed_filename
)
from .models import Index, IndexInstanceNode, DocumentRenameCount
```

All local app module imports are in relative form, local app module name is to be referenced as little as possible, unless required by a specific feature, trick, restriction, ie: Runtime modification of the module's attributes.

Incorrect:

```
# documents app views.py model
from documents.models import Document
```

Correct:

```
# documents app views.py model
from .models import Document
```

Dependencies

Mayan EDMS apps follow a hierarchical model of dependency. Apps import from their parents or siblings, never from their children. Think plugins. A parent app must never assume anything about a possible existing child app. The documents app and the Document model are the basic entities they must never import anything else. The common and main apps are the base apps.

Variables

Naming of variables should follow a Major to Minor convention, usually including the purpose of the variable as the first piece of the name, using underscores as spaces. camelCase is not used in Mayan EDMS.

Examples:

Links:

```
link_document_page_transformation_list = ...
link_document_page_transformation_create = ...
link_document_page_transformation_edit = ...
link_document_page_transformation_delete = ...
```

Constants:

```
PERMISSION_SMART_LINK_VIEW = ...
PERMISSION_SMART_LINK_CREATE = ...
PERMISSION_SMART_LINK_DELETE = ...
PERMISSION_SMART_LINK_EDIT = ...
```

Classes:

```
class Document(models.Model):
class DocumentPage(models.Model):
class DocumentPageTransformation(models.Model):
class DocumentType(models.Model):
class DocumentTypeFilename(models.Model):
```

Strings

Quotation character used in Mayan EDMS for strings is the single quote. Double quote is used for multiple line comments or HTML markup.

Migrations

Migrations should do only one thing (eg: either create a table, move data to a new table or remove an old table) to aid retrying on failure.

General

Code should appear in their modules in alphabetic order or in their order of importance if it makes more sense for the specific application. This makes visual scanning easier on modules with a large number of imports, views or classes. Class methods that return a value should be pretended with a `get_` to differentiate from an object's properties. When a variable refers to a file it should be named as follows:

- filename: The file's name and extension only.
- filepath: The entire path to the file including the filename.
- path: A path to a directory.

Flash messages should end with a period as applicable for the language. Only exception is when the tail of the message contains an exceptions message as passed directly from the exception object.

Source Control

Mayan EDMS source is controlled with [Git](#).

The project is publicly accessible, hosted and can be cloned from [GitLab](#) using:

```
git clone https://gitlab.com/mayan-edms/mayan-edms.git
```

Git branch structure

Mayan EDMS follows a simplified model layout based on Vincent Driessen's [Successful Git Branching Model](#) blog post.

develop The “next release” branch, likely unstable.

master Current production release (2.2).

feature/ Unfinished/unmerged feature.

series/ Released versions.

Each release is tagged separately.

When submitting patches, please place your code in its own `feature/` branch prior to opening a Merge Request on [GitLab](#).

Steps to deploy a development version

```
git clone https://gitlab.com/mayan-edms/mayan-edms.git
cd mayan-edms
git checkout development
virtualenv venv
```

```
source venv/bin/activate
pip install -r requirements.txt
./manage.py initialsetup
./manage.py runserver
```

Setting up a development version using Vagrant

Make sure you have Vagrant and a provider properly installed as per <https://docs.vagrantup.com/v2/installation/index.html>

Start and provision a machine using:

```
vagrant up development
```

To launch a standalone development server

```
vagrant ssh
vagrant@vagrant-ubuntu-trusty-32:~$ cd ~/mayan-edms/
vagrant@vagrant-ubuntu-trusty-32:~$ source venv/bin/activate
vagrant@vagrant-ubuntu-trusty-32:~$ ./manage.py runserver 0.0.0.0:8000
```

To launch a development server with a celery worker and Redis as broker

```
vagrant ssh
vagrant@vagrant-ubuntu-trusty-32:~$ cd ~/mayan-edms/
vagrant@vagrant-ubuntu-trusty-32:~$ source venv/bin/activate
vagrant@vagrant-ubuntu-trusty-32:~$ ./manage.py runserver 0.0.0.0:8000 --
  ↪settings=mayan.settings.celery_redis
```

Then on a separate console launch a celery worker from the same provisioned Vagrant machine:

```
vagrant ssh
vagrant@vagrant-ubuntu-trusty-32:~$ cd ~/mayan-edms/
vagrant@vagrant-ubuntu-trusty-32:~$ source venv/bin/activate
vagrant@vagrant-ubuntu-trusty-32:~$ DJANGO_SETTINGS_MODULE='mayan.settings.celery_
  ↪redis' celery -A mayan worker -l DEBUG -Q checkouts,mailing,uploads,converter,ocr,
  ↪tools,indexing,metadata -Ofair -B
```

Contributing changes

Once you have created and committed some new code or feature, submit a Pull Request. Be sure to merge with the development branch before doing a Pull Request so that patches apply as cleanly as possible. If there are no conflicts, Merge Requests can be merged directly from the website UI otherwise a manual command line merge has to be done and your patches might take longer to get merged.

Debugging

Mayan EDMS makes extensive use of Django's new logging capabilities. By default debug logging for all apps is turned on. If you wish to customize how logging is managed turn off automatic logging by setting *COMMON_AUTO_LOGGING* to False and add the following lines to your `settings/local.py` file:

```
LOGGING = {  
    'version': 1,  
    'disable_existing_loggers': True,  
    'formatters': {  
        'verbose': {  
            'format': '%(levelname)s %(asctime)s %(name)s %(process)d %(thread)d  
↳ %(message)s'  
        },  
        'intermediate': {  
            'format': '%(name)s <% (process) d> [% (levelname) s] "% (funcName) s()  
↳ %(message)s"  
        },  
        'simple': {  
            'format': '%(levelname)s %(message)s'  
        },  
    },  
    'handlers': {  
        'console':{  
            'level':'DEBUG',  
            'class':'logging.StreamHandler',  
            'formatter': 'intermediate'  
        }  
    },  
    'loggers': {  
        'documents': {  
            'handlers':['console'],  
            'propagate': True,  
            'level':'DEBUG',  
        },  
        'common': {  
            'handlers':['console'],  
            'propagate': True,  
            'level':'DEBUG',  
        },  
    }  
}
```

Likewise, to see the debug output of the `tags` app, just add the following inside the `loggers` block:

```
'tags': {  
    'handlers':['console'],  
    'propagate': True,  
    'level':'DEBUG',  
},
```

Documentation

The documentation is written in [reStructured Text](#) format, processed with [Sphinx](#), and resides in the `docs` directory. In order to build it, you will first need to install the documentation editing dependencies with:

```
pip install -r requirements/documentation.txt
```

Then, to build an HTML version of the documentation, run the following command from the `docs` directory:

```
make livehtml
```

The generated documentation can be viewed by browsing to <http://127.0.0.1:8000> or by browsing to the `docs/_build/html` directory.

You can also generate the documentation in formats other than HTML. Consult the [Sphinx](#) documentation for more details.

Installable package

Source file package

This is the sequence of step used to produce an installable package:

1. Make sure there are no lingering packages from previous attempts:

```
rm dist -R
```

2. Generate the packaged version (will produce `dist/mayan-edms-x.y.z.tar.gz`):

```
python setup.py sdist
```

3. Do a test install:

```
cd /tmp
virtualenv venv
source venv/bin/activate
pip install <path of the Git repository>/dist/mayan-edms-x.y.z.tar.gz
mayan-edms.py initialsetup
mayan-edms.py runserver
```

Wheel package

1. Install the development requirements:

```
$ pip install -r requirements/development.txt
```

2. Create wheel package using the source file package (Until issue #99 of wheel is fixed: <https://bitbucket.org/pypa/wheel/issue/99/cannot-exclude-directory>):

```
$ pip wheel --no-index --no-deps --wheel-dir dist dist/mayan-edms-x.y.z.tar.gz
```

3. Do a test install:

```
$ cd /tmp
$ virtualenv venv
$ source venv/bin/activate
$ pip install <path of the Git repository>/dist/mayan_edms-x.y.z-py2-none-any.whl
$ mayan-edms.py initialsetup
$ mayan-edms.py runserver
```

Version numbering

Mayan EDMS uses the Semantic Versioning (<http://semver.org/>) method to choose version numbers along with Python's PEP-0440 (<https://www.python.org/dev/peps/pep-0440/>) to format them.

X.YaN # Alpha release X.YbN # Beta release X.YrcN # Release Candidate X.Y # Final release

CHAPTER 7

App creation

Mayan EDMS apps are essentially Django app with some extra code to register navigation, permissions and other relationships.

App modules

- `__init__.py`

Should be empty if possible. No initialization code should be here, use the `ready()` method of the `Mayan AppConfig` class in the `apps.py` module.

- `admin.py`

Standard Django app module to define how models are to be presented in the admin interface.

- `api_views.py`

REST API views go here. Mayan EDMS uses Django REST Framework API view classes.

- `apps.py`

Contains the `Mayan AppConfig` subclass as required by Django 1.7 and up. This is a place to define the app name and translatable verbose name as well as code to be execute when the modules of the app are ready.

- `classes.py`

Hold python classes to be used internally or externally. Any class defined by the app that is not a model.

- `events.py`

Define event class instances that are later committed to a log by custom code.

- `exceptions.py`

Custom exceptions defined by the app.

- `fields.py`

Place any custom form field classed you define here.

- `forms.py`
Standard Django app module that hold custom form classes.
- `handlers.py`
Contains the signal handlers, functions that will process a given signal emitted from this or other apps. Connect the handler functions to the corresponding signal in the `ready()` method of the `MayanAppConfig` subclass in `apps.py`
- `links.py`
Defines the links to be used by the app. Import only from the navigation app and the local `permissions.py` file.
- `literals.py`
Stores magic numbers, module choices (if static), settings defaults, and constants. Should contain all capital case variables. Must not import from any other module.
- `managers.py`
Standard Django app module that hold custom model managers. These act as model class method to performs actions in a series of model instances or utilitarian actions on external models instances.
- `models.py`
Standard Django app module that defines ORM persistent data schema.
- `permissions.py`
Defines the permissions to be used to validate user access by links and views. Imports only from the `permissions` app. Link or view conditions such as testing for `is_staff` or `is_superuser` flag are defined in this same module.
- `runtime.py`
Use this module when you need the same instance of a class for the entire app. This module acts as a shared memory space for the other modules of the app or other apps.
- `serializers.py`
Hold Django REST Framework serializers used by the `api_views.py` module.
- `settings.py`
Define the configuration settings instances that the app will use.
- `signals.py`
Any custom defined signal goes here.
- `statistics.py`
Provides functions that will compute any sort of statistical information on the app's data.
- `tasks.py`
Code to be execute in the background or as an out-of-process action.
- `tests/ directory`
Hold test modules. There should be one `test_*.py` module for each aspect being tested, examples: `test_api.py`, `test_views.py`, `test_parsers.py`, `test_permissions.py` Any shared constant data used by the tests should be added to `tests/literals.py`
- `utils.py`
Holds utilitarian code that doesn't fit on any other app module or that is used by several modules in the app. Anything used internally by the app that is not a class or a literal (should be as little as possible)

- `widgets.py`

HTML widgets go here. This should be the only place with presentation directives in the app (aside the templates).

Views

The module `common.generics` provides custom generic class based views to be used. The basic views used to create, edit, view and delete objects in Mayan EDMS are: `SingleObjectCreateView`, `SingleObjectDetailView`, `SingleObjectEditView`, and `SingleObjectListView`

These views handle aspects relating to view permissions, object permissions, post action redirection and template context generation.

CHAPTER 8

Roadmap

- Goals for version 2.2:
 - Improve workflow system
 - Workflow indexing support. Accessor already works `{ document.workflows.all.0.get_current_state }`. Index recalculation after workflow transition is missing.
 - Workflow actions. Predefined actions to be execute on document leaving or entering a state or a transition. Example: “Add to folder X”, “Attach tag X”.
 - Add support for state recipients.
 - Add workflow document inbox notification.
 - Replace indexing and smart linking template language (use Jinja2 instead of Django’s).
 - Display/find documents by their current workflow state.
- Goals for version 3.0:
 - Replace UI.
- General goals:
 - Distribution:
 - * Debian packages. Limited success so far using <https://github.com/astraw/stdeb>.
 - Downloads:
 - * Transition from filetransfer package to django-downloadview. This task was started and the view `common.generics.SingleObjectDownloadView` was created. The `document_signatures` app is the first app to use it.
 - Notifications:
 - * Add support for subscribing to a document’s events.
 - * Add support for subscribing to a document type events.
 - * Add support for subscribing specific events.

- OCR:
 - * Add image preprocessing for OCR. Increase effectiveness of Tesseract.
 - * Improve interface with tesseract.
 - * Fix pytesseract shortcomings via upstream patches or re-implement. Move to PyOCR.
- Python 3:
 - * Complete support for Python3.
 - * Find replacement for pdfminer (Python3 support blocker). Use pdfminer.six (#257).
- Simple serving:
 - * Provide option to serve Mayan EDMS without a webserver (using Tornado o similar). Work started in branch: /feature/tornado
- Source code:
 - * Implement Developer certificate of origin: <http://developercertificate.org/>
- Upload wizard:
 - * Make wizard step configurable. Create WizardStep class so apps can add their own upload wizard steps, instead of the steps being hardcoded in the sources app.
 - * Add upload wizard step to add the new documents to a folder.
- Other
 - * Use a sequence and not the document upload date to determine the document version sequence. MySQL doesn't store milisecond value in dates and if several version are uploaded in a single second there is no way to know the order or which one is the latest. This is why the document version tests include a 2 second delay. Possible solution: <http://schinckel.net/2015/05/17/django-second-autofield/>
 - * Include external app Mayan-EXIF into main code.
 - * Convert all views from functions to class based views (CBV).
 - * Increase test coverage.
 - * Mock external services in tests. For example the django_GPG app key search and receive tests.
 - * Pluggable icon app. Make switching icon set easier.
 - * Reduce dependency on binary executables for a default install.
 - * Find replacement for cssmin & django-compressor.
 - * Find replacement for python-gnupg. Unstable & inconsistent API.
 - * Google docs integration. Upload document from Google Drive.
 - * Get dumpdata and loaddata working flawlessly. Will allow for easier backups, restores and database backend migrations.
 - * Make more view asynchronous:
 - trash can emptying.
 - document delete view.
 - * Add support for loading settings from environment variables, not just settings/local.py.
 - * Add generic list ordering. django.views.generic.list.MultipleObjectMixin (<https://docs.djangoproject.com/en/1.8/ref/class-based-views/mixins-multiple-object/#django.views.generic.list.MultipleObjectMixin>) now supports an ordering parameter.

- * Workaround GitLab CI MySQL test errors. GitLab MySQL's container doesn't support UTF-8 content.
- * Add support for downloading the OCR content as a text file.
- * Add support to convert any document to PDF. <https://gitlab.mister-muffin.de/josch/img2pdf>
- * Add support for combining documents.
- * Add support for splitting documents.
- * Add task viewer.
- * Add new document source to get documents from an URL.
- * Document overlay support. Such as watermarks. <https://gist.github.com/umrashrf/8616550>
- * Add support for metadata mapping files. CSV file containing filename to metadata values mapping, useful for bulk upload and migrations.
- * Add support for registering widgets to the home screen.
- * Merge mimetype and converter apps.
- * Add entry in About menu to check latest Mayan EDMS version via PyPI.
- * Add GPG key generation.
- * Add documentation section on editing the settings/local.py file.
- * Add documentation section with warning about using runserver.
- * Replace `urlpatterns = patterns(''`, with Python lists. Django recommendation for post 1.7.
- * If SourceColumn label is None take description from model. Avoid unnecessary translatable strings.
- * Metadata widgets (Date, time, timetdate).
- * Datetime widget: <https://github.com/smalot/bootstrap-datetimepicker>
- * Separate Event class instances with a parent namespace class: EventNamespace.
- * Add events for document signing app (uploaded detached signatue, signed document, deleted signature)
- * A configurable conversion process. Being able to invoke different binaries for file conversion, as opposed to the current libreoffice only solution.
- * A tool in the admin interface to mass (re)convert the files (basically the page count function, but then applied on all documents).
- * Find solution so that documents in watched folders are not processed until they are ready. Use case scanning directly to scanned folders.

CHAPTER 9

Translations

Translations are handled online via the **Transifex** website: <https://www.transifex.com/projects/p/mayan-edms/>. To create a translation team for a new language or contribute to an already existing language translation, create a **Transifex** account and contact the team coordinator of the respective language in which you are interested.

Feel free to open translation issues inside **Transifex** itself if you have a question about the usage or meaning of a source text string. If you open a translation issue, it will be your responsibility to close it after you get an answers that satisfies your question. Administrator will not close new issues as they have no way to determine if your question has been properly answered. However to avoid clutter, answered questions will be scanned periodically and closed if no activity is observed from the original poster in a period of time.

CHAPTER 10

Contributors

How to contribute?

You can help further the development of Mayan EDMS by testing, reporting bugs, submitting documentation or code patches.

Lead developer

- Roberto Rosario (roberto.rosario@mayan-edms.com) <<https://twitter.com/siloraptor>>

Contributors (in alphabetical order)

- Aziz M. Bookwala (<https://github.com/azizmb>)
- Bertrand Bordage (<https://github.com/BertrandBordage>)
- Brian E (brian@realize.org)
- David Herring (<https://github.com/abadger1406>)
- Emlyn Clay (<https://github.com/EmlynC>)
- Jens Kadenbach (<https://github.com/audax>)
- Kolmar Kafran
- Helga Carrero
- IHLeanne (<https://github.com/IHLeanne>)
- Iliya Georgiev (ikgeorgiev@gmail.com)
- Lars Kruse (devel@sumpfralle.de)

- Mathias Behrle <mbehrle@m9s.biz>
- Meurig Freeman (<https://github.com/meurig>)
- Nate Aune (nate@appsembler.com)
- Paul Whipp [<https://github.com/pwhipp>] [<http://paulwhippconsulting.com>]
- Rafael Esparra <rafael.esparra1@upr.edu>
- Sergey Glita (s.v.glita@gmail.com)
- Simone Federici [<https://twitter.com/aldaranalton>] [<https://github.com/simone>]
- Webfaction (<https://www.webfaction.com>)

CHAPTER 11

License

Mayan EDMS is distributed under the [Apache 2.0 License](#). The complete license terms are included below.

License terms

Copyright 2011-2015 Roberto Rosario

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.

CHAPTER 12

FAQ

Frequently asked questions and solutions

Q: PostgreSQL vs. MySQL

Since Django abstracts database operations from a functional point of view Mayan EDMS will behave exactly the same either way. The only concern would be that MySQL doesn't support transactions for schema modifying commands. The only moment this could cause problems is when running South migrations during upgrades, if a migration fails the database structure is left in a transitory state and has to be reverted manually before trying again.

Q: `_mysql_exceptions.OperationalError: (1267, "Illegal mix of collations (latin1_swedish_ci, IMPLICIT) and (utf8_general_ci, COERCIBLE) for operation '='")`

- Solution:

```
$ mayan-edms.py shell
```

```
>>> from django.db import connection
>>> cursor = connection.cursor()
>>> cursor.execute('SHOW TABLES')
>>> results=[]
>>> for row in cursor.fetchall(): results.append(row)
>>> for row in results: cursor.execute('ALTER TABLE %s CONVERT TO CHARACTER SET
    ↴utf8 COLLATE utf8_general_ci;' % (row[0]))
```

- References:

- <http://www.djangoshmango.com/?p=99>
- <http://stackoverflow.com/questions/1073295/django-character-set-with-mysql-weirdness>

Q: Incorrect string value: “xE2x80x95rs6...”“ for column “content” at row 1

When using MySQL and doing OCR on languages other than English

- Solution:

- Use utf-8 collation on MySQL server, or at least in table ‘documents_documentpage’, ‘content’ field

- Ref: 1- http://groups.google.com/group/django-users/browse_thread/thread/429447086fca6412
- Ref: 2- <http://markmail.org/message/bqajx2utvmtriixi>

Q: Error “django.db.utils.IntegrityError IntegrityError: (1452, ‘Cannot add or update a child row: a foreign key constraint fails (...:..., CONSTRAINT ..._refs_id_b0252274 FOREIGN KEY (...) REFERENCES ... (...))’)

- Solution:
 - Convert all MySQL tables to the same type, either all MyISAM or InnoDB

Q: File system links not showing when serving content with “Samba”

- Solution:
 - Disable unix extensions in the [global] section and enable wide links for the file serving share
 - Example:

```
[global]
unix extensions = no

...
[digitalizacion]
path = /var/local/mayan
guest ok = yes
read only = yes
wide links = yes
follow symlinks = yes
```

- Ref: 1- <http://www.samba.org/samba/docs/man/manpages-3/smb.conf.5.html>

Q: How do you upload a new version of an existing file?

- Solution:
 - Choose a document, and go to the versions tab, on the right menu at the bottom under Other available action there is Upload new version. Clicking it will take you to a very similar view as the Upload new document but you will be able to specify version number and comments for the new version being uploaded.

Q: Is virtualenv required as specified in the documentation?

- It is not necessary, but it's a strong recommendation mainly to reduce dependency conflicts by isolation from the main Python system install. If not using a virtualenv, pip would install Mayan's dependencies globally coming in conflict with the distribution's prepackaged Python libraries messing other Django projects or Python programs, or another later Python/Django project dependencies coming into conflict causing Mayan to stop working for no apparent reason.

Q: Mayan EDMS installed correctly and works, but static files are not served

Django's development server doesn't serve static files unless the DEBUG option is set to True, this mode of operation should only be used for development or testing. For production deployments the management command:

```
$ mayan-edms.py collectstatic
```

should be used and the resulting static folder served from a webserver. For more information, read <https://docs.djangoproject.com/en/dev/howto/static-files/> and <https://docs.djangoproject.com/en/1.2/howto/static-files/> or http://mayan-edms-ru.blogspot.com/2011/11/blog-post_09.html

Q: Can you change the display order of documents...i.e can they be in alphabetical order?

A the moment no, but it is something being considered.

Q: Does Mayan EDMS work with Python 3?

Not at the moment. When all the projects and libraries upon which Mayan is dependent support Python 3 then will the project move to fully support Python 3.

Q: The watched folder feature is not working

Make sure that the Celery BEAT scheduler is running correctly as it is the element that triggers the periodics tasks.

CHAPTER 13

Contact

FAQ

- Try the [FAQ](#) – it's got answers to many common questions.

Mailing list

Search for information in the [archives](#) of the mayan-edms mailing list, or [post a question](#). If you prefer news servers, use the gateway provided by [Gmane](#).

Mayan EDMS community developers do their best to reply to basic questions. Be sure to check the list archives as it may already contain the answers to your questions.

Twitter

Mayan EDMS has an official Twitter account, [@mayanedms](#), which is used for announcements and occasional related news tidbits.

Bugs/ticket tracker

Report bugs with Mayan EDMS or search existing ones using Gitlab's [ticket tracker](#).