

{appe}

Table of contents

Table of contents

[Introduction](#)

[How to configure](#)

[Source files and tools](#)

[app. functions and hooks](#)

Recipes

[Customize: rent cars to dinosaurs](#)

[Extend: export to PDF](#)

Introduction

The primary intent was to allow simple data manipulation without have to install web servers, tons of software and deps, create a portable app quickly with the possibility to write *spaghetti code*, fairly simple to use, executable almost anywhere, with essential features and small footprint.

Folder structure

The folder structure is simplified to facilitate the user, there is a launcher file to open "*LAUNCH.html*", all the runtime content is inside the folder "*app*", save files are contained in the folder "*save*", extensions in the folder "*ext*", the alternate execution in the folder "*alt*". In the release phase you could delete superfluous files and folders like "docs" and "src".

- alt
- app
 - assets
 - views
 - about.html
 - sample-edit.html
 - sample-wide-list.html
 - sample.html
 - config.js
 - index.html
 - LICENSE.txt
 - README.md
 - robots.txt
- ext
- save
- LAUNCH.html

How it works

The source JavaScript is located in "*src/app/src*", it is divided into sections to simplify understanding, the name of sections is intuitive.

The index file "*index.html*" contained in the runtime folder "*app*" acts as a router and main controller, it reflects the functions contained in "*main.js*".

All views, which correspond to routes and events declared in the configuration file "*config.js*" are contained in the "*views*" folder, they reflect the functions contained in "*view.js*".

The application is started from the launcher file "[LAUNCH.html](#)"; if it has been previously executed and the configuration foresees the saving in JavaScript file (.js) it will try to resume first the previous session, alternatively will attempt to reload the last file loaded from the save folder "[save](#)". Then you are redirected to the index file "[index.html](#)" which will manage all directions in the app.

The storage into the application is provided by "[localStorage](#)" and "[sessionStorage](#)" browser APIs.

To appreciate features and functionality of individual functions read the documentation section [app. functions and hooks](#).

Further details on customization and extensibility of **appe** are in the section [Recipes](#).

How to configure

Read the comments in the global *"appe_config"* object.

```

appe__config = {
  "app_ns": "your_app_namespace", // the namespace of the app (required)
  "launcher_name": "LAUNCH", // the name of the launcher (required)
  "app_name": "your app name", // the name of the app (required)
  "language": "en", // language, set to null for auto-select (required)
  "language_direction": "ltr", // language direction, accepted values are "ltr" or "rtl", default to "ltr"
  "compression": false, // session file compression
  "encryption": true, // session file encryption
  "binary": true, // saves session to binary file
  "secret_passphrase": "test", // the secret passphrase needed with session file compression active
  "verify_checksum": true, // whenever to verify JSON file checksum
  "debug": true, // turns on debug
  "schema": [ // the file schema (required)
    "file",
    ...
  ],
  "events": { // app events, { "event name": "event slug", ... } (required)
    "event-name": "event-slug",
    ...
  },
  "routes": { // app routes, { "parent route slug/filename": { "child route event slug": "child route filename"
    , ... }, ... } (required)
    "parent-route-filename": {
      "child-route-event-slug": "child-route-filename",
      "child-route-event-slug-edit": "child-route-filename-edit",
      "child-route-event-slug-whole": "child-route-filename-whole",
      ...
    },
    ...
  },
  "default_route": "parent-route-filename", // default route (required)
  "default_event": "event", // default event (required)
  "base_path": "appe", // where is located the root folder (required)
  "runtime_path": "app", // where is located the runtime folder (required)
  "save_path": "save", // where is located the save folder (required)
  "aux_path": "ext", // where is located the extensions folder
  "load_attempts": 50, // number of attempts to resume file session or load extensions, default to 50 for best
  effort
  "alt": { // alternate execution, used when occurring browser or system limitations, it is a fallback
    "exec_folder": "alt", // where is located the alternate folder
    "exec_platform": { // system specifics
      "win": "LAUNCH.win.exe",
      "mac": "LAUNCH.mac.app",
      "hta": "LAUNCH.hta"
    }
  }
  "aux": [ // extensions, asynchronous loaded
    {
      "file": "filename.js", // extension filename (required)
      "fn": "extension_global_scope", // the global name to declare inside the extension, used to verify load a
      nd expose features (required)
      "memo": false // stores the extension exposed features into app global store
    }
  ],
  "license": { // license app related
    "text": "LGPL-3.0-or-later", // textile license
    "file": "LICENSE.txt" // file license
  },
  "file": { // session file settings
    "binary": true,
    "compress": true,
    "filename_prefix": "appe_save",
    "filename_separator": "_",
    "filename_date_format": "Y-m-d_H-M-S",
  },
  "csv": { // csv export settings
    "filename_prefix": "csv_export",
    "filename_separator": "_",
    "filename_date_format": "Y-m-d_H-M-S",
  },
}

```

Source files and tools

The source JavaScript is located in "`src/apppe/src`", it is divided by sections to simplify understanding, the name of sections is intuitive.

- **controller.js** (Controller functions)
- **index.js** (Index with most important functions)
- **layout.js** (Handles layout functions)
- **lockdown.js** (Object.freeze)
- **main.js** ("main" functions)
- **memory.js** (Handles storage entries)
- **os.js** (Handles filesystem functions)
- **start.js** (Launcher functions)
- **store.js** (Handles persistent storage entries)
- **utils.js** (Utils functions)
- **view.js** ("view" functions)

Scripting tools

Scripts available in **appe** via `npm run-script`:

Task	Description
watch	build/watch all js and css assets
watch-js	build/watch all js assets
watch-css	build/watch all css assets
build	build all js and css assets
build-js	build all js assets
build-css	build all css assets
build-docs	build all documentation files, shortcut to <code>php generate_docs.php</code>
build-docs-html	transform wiki markdown documentation in html
build-docs-pdf	transform wiki markdown documentation in pdf
copy-demo	recursively copy all app assets in "demo" folder
copy-docs-recipes	recursively copy copy all documentation recipes files in "docs" folder
copy-electron	recursively copy all app files in " <code>src/electron/src</code> " folder

Electron (as an alternate execution)

To permits **apfe** to run when the browser or the system do not allow direct execution, is used [Electron](#), you can use the strategy that suits your needs.

The package configuration and the necessary dependencies are provided by [Electron Forge](#), to install it `npm install -g electron-forge`.

After installing the necessary dependencies from the "[src/electron](#)" folder with the `electron-forge start electron` command, you can create distribution packages for operating systems with the `electron-forge package` command.

All the informations for package build are available on the website <https://electronforge.io>.

app. functions and hooks

app

- load
- unload
- beforeunload
- position
- session
- resume
- redirect
- data
- checkConfig
- checkFile
- openSessionFile
- saveSessionFile
- newSession
- openSession
- saveSession
- asyncAttemptLoad
- asyncLoadAux
- i18n
- debug
- stop
- error
- blind
- getInfo
- getName
- getVersion
- getLicense
- getLocale

app.os

- fileSessionOpen
- fileSessionSave
- fileDownload
- fileFindRoot
- scriptOpen
- generateFileHead
- generateJsonChecksum
- getLastFileName
- getLastFileVersion
- getLastFileChecksum
- getLastFileHead

app.controller

- spoof
- history
- cursor
- setTitle
- getTitle
- store
- retrieve
- clear

app.memory

- set
- get
- has
- del
- reset

app.store

- set
- get
- has

- [del](#)
- [reset](#)

app.start

- [redirect](#)
- [alternative](#)
- [load](#)
- [progress](#)
- [loadComplete](#)

app.main

- [control](#)
- [handle](#)
- [action](#)
- [load](#)
- [beforeunload](#)
- [loadComplete](#)
- [setup](#)

app.view

- [spoof](#)
- [control](#)
- [action](#)
- [sub](#)
- [handle](#)
- [send](#)
- [fetch](#)
- [resize](#)
- [getFormData](#)
- [convertTableCSV](#)
- [copyToClipboard](#)
- [load](#)
- [beforeunload](#)

- loadComplete

app.layout

- renderElement
- renderSelect
- renderSelectOption
- renderSelectOptionGroup
- renderSelectOptions
- dropdown
- collapse
- draggable
- localize

app.utils

- system
- addEvent
- removeEvent
- proxy
- storage
- cookie
- base64
- transform
- sanitize
- classify
- numberFormat
- dateFormat
- numberLendingZero
- isPlainObject
- extendObject

Hooks

- start.session
- main.session

- `control.session`
- `start.loadComplete`
- `start.alternative`
- `main.handle`
- `main.setup`
- `control.renderRow`
- `control.fillForm`
- `control.openView`
- `control.loadComplete`

Customize: rent cars to dinosaurs

1. Modify the configuration

This is an example of customization starting from the sample event "events.custom"_ in the demo file "[sample.html](#)".

First you need to edit the file "[config.js](#)", add a new event and a new route, then create the page to serve with the new view.

```
appe__config = {  
  
  "events": {  
    "selection": "selection",  
    "update": "update",  
    "open": "open",  
    "close": "close",  
    "add": "add",  
    "edit": "edit",  
    "delete": "delete",  
    "list": "list",  
  
    "rent": "rent" /* <== */  
  
  },  
  
  "sample": {  
    "add": "sample-edit",  
    "edit": "sample-edit",  
    "delete": "sample",  
  
    "rent": "sample-rent" /* <== */  
  
  },  
  
}
```

2. Add the custom event to "index.html"

To create an event with a route to be served, it must be declared in the index file "[index.html](#)".

Therefore add an alias "rent" to the function "[app.main.handle.prototype.prepare](#)", that is a generic method performing all necessary actions to set action, route and update data.

```

main.handle = function(handler, event, ctl) {
  if (! handler || ! event || ! ctl) {
    return app.error('main.handle', arguments);
  }

  var _events = app.utils.extendObject({}, handler);

  _events.selection = function() {
    try {
      var _data = JSON.parse(ctl.data);

      app.memory.set('archive_id', parseInt(_data.id));
    } catch {
      return;
    }

    return handler.selection();
  }

  _events.rent = _events.prepare; // alias to "prepare" aka "app.main.handle.prototype.prepare"

  _events[event]();
}

```

3. Create the view "sample-rent.html"

Once the configuration has been modified, create the file with the view, it can be used as model *"sample-edit.html"* which is closer to the needs.

I used a *"rent"* object grafted into the parent item *"items"*, which pre-existed in the scheme, for convenience.

The structure of the view is almost identical to the reference model *"edit"*, also in this case the *"rent"* event is declared as an alias of the function *"app.view.action.prototype.prepare"*.

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" type="text/css" href="../assets/css/lib/bootstrap/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="../assets/css/app.min.css">
</head>

<body class="view view_edit">
<h1 id="view-title" class="page-header">Items</h1>
<h2 id="section-title" class="sub-header"></h2>

<div id="section-actions-top" class="section-actions section-actions-top">
  <div>
    <a class="btn btn-sm btn-gray-lighter" href="../index.html?sample" target="_parent">&laquo; Return Back</a>
  </div>
</div>

<form id="form-data" class="form form-data" autocomplete="off" validate>
<fieldset>
<div class="row">
<div class="form-group col-6 col-lg-4">
  <label for="sample_rent_dino">Dinosaur</label>
  <select id="sample_rent_dino" name="rent[dino]" class="form-control" required></select>

```

```

<select id="sample_rent_unit" name="rent[unit]" class="form-control required"></select>
</div>
</fieldset>

<fieldset>
<legend>Rent</legend>

<div class="row">
<div class="form-group col-12 col-lg-3">
<label for="sample_rent_period">Period</label>
<select id="sample_rent_period" name="rent[period]" class="form-control">
<option value="1">1 month</option>
<option value="2">2 months</option>
<option value="3">3 months</option>
<option value="4">4 months</option>
</select>
</div>

<div class="clearfix visible-lg-block"></div>

<div class="form-group col-12 col-lg-2">
<label for="sample_rent_amount">Amount</label>
<div class="input-group">
<input type="text" id="sample_rent_amount" name="rent[amount]" class="form-control" value="0.00" data-trans
form="numeric" required>
<div class="input-group-addon">
<span class="input-group-text">coin &times; month</span>
</div>
</div>
</div>
</div>
</fieldset>

<fieldset class="form-submit">
<input type="hidden" id="index" name="id" value="{id}" data-transform="integer">
<input type="submit" id="real-submit" class="hide" hidden>
<button type="button" id="submit" class="btn btn-primary btn-lg" onclick="control.action(this, 'edit')">Rent
Vehicle</button>
</fieldset>
</form>

<script type="text/javascript" src="../../assets/js/app.min.js"></script>

<script type="text/javascript" src="../../config.js"></script>

<script type="text/javascript">
var control = appe__control = {};

control.handle = function(data) {
var data_sample = data['sample'];
var data_items = data['items'];

var form = document.getElementById('form-data');

if (! data_sample || ! data_items) {
return app.error('control.handle', 'data*');
}

var _control = app.view.control([ 'rent' ], data_sample, form);

_control.begin();

var event = _control.getEvent();
var id = _control.getID();

```



```

var _events = {};

_events.rent = function() {
  _control.setTitle('Rent Vehicle');

  _control.setActionHandler();

  control.fillForm(data_sample[id]);
}

var dino_selection = [ { "0": "&ndash; select &ndash;" } ];

Array.prototype.forEach.call(Object.keys(data_items), function(id) {
  var obj = {};
  obj[id] = data_items[id].name;
  dino_selection.push(obj);
});

var dino_selected = data_sample[id].rent ? data_sample[id].rent[dino] : 0;

var select_dino = document.getElementById('sample_rent_dino');
select_dino.innerHTML = app.layout.renderSelectOptions('sample_rent_dino', dino_selection, dino_selected);
select_dino.value = dino_selected;

_events[event]();

_control.end();
}

control.fillForm = function(data_field) {
  if (! data_field) {
    return app.error('control.fillForm', arguments);
  }

  data_field.rent = data_field.rent || { dino: 0, period: 1, amount: '0.00' }; // to prevent is not defined error, fill with empty data

  document.querySelector('#sample_rent_dino > option[value="' + data_field.rent.dino + '"').setAttribute('selected', '');
  document.getElementById('sample_rent_period').setAttribute('value', data_field.rent.period);
  document.getElementById('sample_rent_amount').setAttribute('value', data_field.rent.amount);
}

control.action = function(element, event, extra) {
  var form = document.getElementById('form-data');

  if (! element || typeof event !== 'string') {
    return app.error('control.action', arguments);
  }

  var _action = app.view.action([ 'rent' ], event, element, form);

  _action.begin();

  _action.rent = _action.prepare; // alias to "prepare" aka "app.view.action.prototype.prepare"

  if (_action.validateForm()) {
    return;
  }
}

```

```

var id = _action.getID();

var _data_updated = {};

var _events = {};

_events._prepare = function() {
    var _ext_data = app.utils.extendObject(true, _data_updated['sample'][id], app.view.getFormData(form.elements));
    _data_updated['sample'][id] = _ext_data;
}

_events.rent = function() {
    var data_sample = app.data('sample');

    if (! data_sample) {
        return app.error('control.action().rent', 'data*');
    }

    _data_updated['sample'] = data_sample;

    _events._prepare();

    return _action.rent(_data_updated, true);
}

var ctl = _events[event](extra);

app.view.send(ctl);

_action.end();
}

app.load(app.view.load);
app.beforeunload(app.view.beforeunload);
</script>
</body>
</html>

```

4. Modify the view "sample.html"

Finally all changes introduced in the new view *"sample-rent.html"* will be integrated into the existing view *"sample.html"*.

Rename the *"example"* event with *"rent"*, in the links and in all the control functions.

```

<table id="table-data" class="table table-data">
  <thead>
    <tr>
      <th class="hidden-print">Actions</th>
      <th>ID</th>
      <th class="text-nowrap">Vehicle Name</th>
      <th class="text-nowrap">Vehicle Type</th>
      <th class="text-nowrap">Manufactured</th>
      <th class="text-nowrap">Value Price</th>

      <th>Rent Dino</th>
      <th>Rent Period</th>
      <th>Rent Amount</th>

      <th class="hidden-print">Rent</th>
    </tr>
  </thead>
  <tbody>
    <tr class="tpl">
      <td class="hidden-print">
        <a class="action btn btn-secondary" href="../../index.html?sample&edit={id}" onclick="control.action(this, 'edit'); return false;" target="_parent">Edit</a>
        <a class="action btn btn-danger" href="../../index.html?sample&delete={id}" onclick="control.action(this, 'delete'); return false;" target="_parent">Delete</a>
      </td>
      <td>{id}</td>
      <td>{title}</td>
      <td>{type}</td>
      <td class="text-nowrap">{date}</td>
      <td class="text-nowrap">{amount}</td>

      <td>{rent_dino}</td>
      <td>{rent_period}</td>
      <td>{rent_amount}</td>

      <td class="hidden-print">

        <a class="action btn btn-link btn-sm" href="../../index.html?sample&rent={id}" onclick="control.action(this, 'rent'); return false;" target="_parent">RENT</a>

      </td>
    </tr>
  </tbody>
</table>

```

```

control.action = function(element, event, extra) {
  var data = app.data();

  var data_sample = data['sample'];

  var table = document.getElementById('table-data');

  if (! data_sample) {
    return app.error('control.action', arguments);
  }

  if (! element || typeof event !== 'string') {
    return app.error('control.action', arguments);
  }

  var _action = app.view.action([ 'edit', 'delete', 'export', 'print', /* ==> */ 'rent' /* <== */ ], event, element, data_sample);

  _action.begin();

  var id = _action.getID();

  var _data_updated = {};

  var _events = {};

```

The *"items"* data source must also be added in order to return the name of the dinosaur, because only the ID will be stored in the *"rent"* object.

```

control.handle = function(data) {
  var data_sample = data['sample'];
  var data_items = data['items']; /* <== */

  var table = document.getElementById('table-data');

  if (! data_sample || ! data_items) { /* <== */
    return app.error('control.handle', 'data*');
  }

  var _control = app.view.control(null, data_sample);

  _control.begin();

  _control.fillTable(table, null, null, data_items); /* <== */ // passing down "data_items" to extra argument

  _control.end();
}

```

All new datas must be integrated in order to render them in the table recognized by the *"table-data"* identifier.

```

control.renderRow = function(tpl_row, id_row, data_row, extra) {
  if (! tpl_row || ! data_row) {
    return app.error('control.renderRow', arguments);
  }

  var items_extra = extra[0]; /* <== */

  var row = tpl_row.cloneNode(true);

  row.setAttribute('data-index', id_row);

  data_row.rent = data_row.rent || { dino: 0, period: 1, amount: 0 }; /* <== */ // to prevent is not defined
  error, fill with empty data

  row.innerHTML = row.innerHTML
    .replace(/\{id\}/g, data_row.id)
    .replace('{title}', data_row.title)
    .replace('{type}', data_row.type)
    .replace('{date}', app.utils.dateFormat(data_row.date, 'd/m/Y'))
    .replace('{amount}', app.utils.numberFormat(data_row.amount, 2, ',', '.') + ' coin')
    .replace('{status}', data_row.status)

    .replace('{rent_dino}', data_row.rent.dino ? items_extra[data_row.rent.dino].name : '&ndash;') /* <== */
    .replace('{rent_period}', data_row.rent.period + ' month') /* <== */
    .replace('{rent_amount}', app.utils.numberFormat(data_row.rent.amount, 2, ',', '.') + ' coin &times; month'
  ); /* <== */

  return row;
}

```

Then the action "event.example"_ must be replaced to change route when the user clicks the RENT button.

```

// Custom action example
_events.rent = function() {
  _action.rent = _action.prepare; // alias to "prepare" aka "app.view.action.prototype.prepare"

  return _action.rent();
}

```

The files in this example are contained in the folder "[docs/recipes/customize](#)".

Extend: export to PDF

1. Write the extension function

To add new functions in **appe** or extend the existing ones create extensions, as an example an extension to extend lists in PDF.

Create a file inside the "ext" extension folder and set up a function with the steps necessary to show the new feature in the views.

The function must be run to make available the new feature in the application.

```
export_to_pdf = function() {
  var _is_view = ! (window.appe__control === undefined);

  if (_is_view) {
    var dropdown = document.querySelector('#section-actions-top .dropdown-menu');

    var item = document.createElement('li');
    var link = document.createElement('a');

    link.setAttribute('href', 'javascript:');
    link.setAttribute('onclick', 'control.action(this, \'export\', \'pdf\'); return false;');
    link.innerHTML = 'Export PDF';

    item.append(link);
    dropdown.append(item);
  }
}

export_to_pdf();
```

2. Modify the configuration

Once the function has been created, edit the file "*config.js*", add the references to the file and the function just created to the "aux" object, the file will be loaded asynchronously when pages loading.

```
appe__config = {

  "aux": [
    {
      "file": "demo.min.js",
      "fn": "appe__demo"
    },
    {
      "file": "export-pdf.js", /* <== */
      "fn": "export_to_pdf" /* <== */
    }
  ]
}
```

3. Add the feature

It is time to code, in this case you can use a pre-existing function as model "*app.view.sub.prototype.csv*" that is closest to the needs.

To achieve your goal you can use JavaScript libraries or write, create functions, etc. .

Create a "*export_to_pdf_view_sub_pdf*" function to reference then within the main "*export_to_pdf*".

```
export_to_pdf_view_sub_pdf = function(element, table) {
  if (! jsPDF) {
    return console.warn('aux: export_to_pdf', '\t', 'jsPDF');
  }

  if (! element || ! table) {
    return console.error('aux: export_to_pdf', '\t', [element, table]);
  }

  var source;
  var table_csv = app.view.convertTableCSV(table);

  var doc = new jsPDF();

  var doc_head = table_csv[0];
  var doc_body = table_csv.slice(1);

  doc.autoTable({ head: [doc_head], body: doc_body });

  var file;

  var filename_prefix = 'pdf_export';
  var filename_separator = '_';
  var filename_date_format = 'Y-m-d_H-M-S';

  var filename = filename_prefix;
  var filename_date = app.utils.dateFormat(true, filename_date_format);

  filename += filename_separator + filename_date;

  file = filename + '.pdf';

  doc.save(file);
}
```

4. Finalize

To complete and show the new feature you need to satisfy all dependencies, this example requires the library *jsPDF* and the plugin *jspdf-autotable*.

A function "*app.view.sub.prototype.pdf*" must be declared as an alias of the one just created "*export_to_pdf_view_sub_pdf*", which extends the prototype function "*app.view.sub.prototype*".

```

export_to_pdf = function() {
  var _is_view = ! (window.appe__control === undefined);

  app.view.sub.prototype.pdf = export_to_pdf_view_sub_pdf;  /* <== */  // alias

  if (_is_view) {
    var dropdown = document.querySelector('#section-actions-top .dropdown-menu');

    var item = document.createElement('li');

    var link = document.createElement('a');
    link.setAttribute('href', 'javascript:');
    link.setAttribute('onclick', 'control.action(this, \'export\', \'pdf\'); return false;');
    link.innerHTML = 'Export PDF';

    item.append(link);
    dropdown.append(item);

    // load required libs

    var head = document.getElementsByTagName('head')[0];
    var script;

    script = document.createElement('script');
    script.setAttribute('src', '../..ext/lib/jspdf/jspdf.min.js');

    head.append(script);

    var _defer = setTimeout(function() {
      script = document.createElement('script');
      script.setAttribute('src', '../..ext/lib/jspdf-autotable/jspdf.plugin.autotable.min.js');

      head.append(script);

      clearTimeout(_defer);
    }, 1000);
  }
}

```

The files in this example are contained in the folder "[docs/recipes/extend](#)".

app.index

app.load

Helper app load function DOM

```
<Function>
```

arguments:

```
<Function> func  
<Boolean>
```

returns.

position:

- [src/appe/src/js/index.js](#) line: 49

app.unload

Helper app unload function DOM

```
<Function>
```

arguments:

```
<Function> func
```

returns.

position:

- [src/appe/src/js/index.js](#) line: 90

app.beforeunload

Helper app before unload function DOM

```
<Function>
```

arguments:

```
<Function> func
```

returns.

position:

- [src/appe/src/js/index.js line: 113](#)

app.position

Returns JSON serialized app position

```
<Function>
```

returns:

```
<String> position
```

position:

- [src/appe/src/js/index.js line: 135](#)

app.session

Initializes the session, returns to callback

```
<Function>    // asynchronous
```

globals:

```
<Object> appe__store  
<Object> appe__locale  
<Object> CryptoJS  
<Object> pako
```

arguments:

```
<Function> callback  
<Object> config  
<String> target
```

returns.

position:

- [src/appe/src/js/index.js line: 170](#)

app.resume

Resumes session, returns last opened file

```
<Function>
```

arguments:

```
<Object> config
<Boolean> target
```

returns:

```
<String> session_resume
```

position:

- [src/appe/src/js/index.js line: 458](#)

app.redirect

Performs app redirect

```
<Function>
```

globals:

```
<Object> appe__config
```

returns.

position:

- [src/appe/src/js/index.js line: 557](#)

app.data

Gets data store

```
<Function>
```

globals:

```
<Object> appe__store
```

arguments:

```
<String> key
```

returns:

```
<Object>
```

position:

- [src/appe/src/js/index.js line: 587](#)

app.checkConfig

Verifies config file

<Function>

arguments:

<Object> config

returns:

<Boolean>

position:

- [src/appe/src/js/index.js](#) line: 620

app.checkFile

Verifies opened file

<Function>

globals:

<Object> appe__config

arguments:

<Object> source
<Object> checksum

returns:

<Boolean>

position:

- [src/appe/src/js/index.js](#) line: 663

app.openSessionFile

Opens session from an app session file

<Function>

globals:

```
<Object>  appe__config
<Object>  appe__start
<Object>  CryptoJS
<Function> pako
```

arguments:

```
<Object>  source
```

returns.

position:

- [src/appe/src/js/index.js](#) line: 723

app.saveSessionFile

Saves session to app session file

```
<Function>
```

globals:

```
<Object>  appe__config
<Object>  appe__store
<Object>  appe__start
<Object>  CryptoJS
<Function> pako
```

arguments:

```
<Object>  source
```

position:

- [src/appe/src/js/index.js](#) line: 842

app.newSession

Creates a new empty session

```
<Function>
```

globals:

```
<Object>  appe__config
<Object>  appe__start
```

returns.

position:

- [src/appe/src/js/index.js](#) line: 904

app.openSession

Opens session, alias of app.openSessionFile

```
<Function>
```

position:

- [src/appe/src/js/index.js](#) line: 971

app.saveSession

Saves session, alias of app.saveSessionFile

```
<Function>
```

position:

- [src/appe/src/js/index.js](#) line: 979

app.asyncAttemptLoad

Attempts to load files and scripts, returns to callback

```
<Function>    // asynchronous
```

globals:

```
<Object>  appe__config  
<Object>  CryptoJS
```

arguments:

```
<Function> callback  
<Boolean> resume_session  
<String>  fn  
<String>  file  
<Object>  schema  
<Boolean> memoize
```

returns.

position:

- [src/appe/src/js/index.js](#) line: 997

app.asyncLoadAux

Load extension scripts asynchronously, returns to callback

```
<Function> // asynchronous
```

globals:

```
<Object> appe__config
```

arguments:

```
<Function> callback  
<Object> routine  
<Boolean> resume_session
```

returns.

position:

- [src/appe/src/js/index.js](#) line: 1168

app.i18n

App localization

TODO: *implement nest replacement*

```
<Function>
```

globals:

```
<Object> appe__locale
```

arguments:

```
<String> to_translate  
<String> context  
to_replace
```

position:

- [src/appe/src/js/index.js](#) line: 1221

app.debug

Utility debug

```
<Function>
```

arguments:

<Object> source

returns:

<Boolean>

position:

- [src/appe/src/js/index.js line: 1348](#)

app.stop

Stops the app execution

<Function>

globals:

<Object> appe__main

arguments:

```
<String> arg0    ( msg | fn )  
arg1    ( log | msg )  
arg2    ( log | msg )  
<Boolean> soft
```

returns:

<Boolean>

position:

- [src/appe/src/js/index.js line: 1365](#)

app.error

Helper to debug and display error messages

<Function>

globals:

<Object> appe__control

arguments:


```
<String> arg0    fn  
arg1    ( log | msg )  
arg2    ( log | msg )  
<Boolean> soft
```

returns:

```
<undefined>
```

position:

- [src/appe/src/js/index.js](#) line: 1400

app.blind

Helper to freeze "main" screen

```
<Function>
```

globals:

```
<Object>  appe__main
```

returns.

position:

- [src/appe/src/js/index.js](#) line: 1466

app.getInfo

Utility to get app info(s)

```
<Function>
```

globals:

```
<Object>  appe__config
```

arguments:

```
<String> from    ( config | runtime )  
<String> info    { config { app_name | schema | license } } | runtime { { debug | locale | version | release } }
```

returns.

position:

- [src/appe/src/js/index.js](#) line: 1490

app.getName

Gets app name

```
<Function>
```

returns:

```
<String>
```

position:

- [src/appe/src/js/index.js](#) line: 1538

app.getVersion

Gets app version

```
<Function>
```

arguments:

```
<String> info
```

returns.

position:

- [src/appe/src/js/index.js](#) line: 1551

app.getLicense

Gets app license

```
<Function>
```

returns.

position:

- [src/appe/src/js/index.js](#) line: 1574

app.getLocale

Gets app locale language

<Function>

returns:

<String>

position:

- [src/appe/src/js/index.js](#) line: 1586

app.os

app.os

Handles filesystem functions

```
<Object>
```

app.os.fileSessionOpen

Opens a session file through the browser FileReader API, stores it, returns to callback

```
<Function> // asynchronous
```

globals:

```
<Object> appe__config  
<Object> CryptoJS  
<Function> pako
```

arguments:

```
<Function> callback
```

returns.

position:

- [src/appe/src/js/os.js](#) line: 20

app.os.fileSessionSave

Sends a file to the browser, returns to callback

```
<Function> // asynchronous
```

globals:

```
<Object> appe__config  
<Object> CryptoJS  
<Function> pako
```

arguments:

```
<Function> callback  
<Object> source  
<Date> timestamp
```

returns.

position:

- [src/appe/src/js/os.js](#) line: 297

app.os.fileDownload

Prepares attachment data file and sends it to browser

links: <https://github.com/eligrey/FileSaver.js/>

```
<Function>
```

arguments:

```
source  
<String> filename  
<String> mime_type
```

returns.

position:

- [src/appe/src/js/os.js](#) line: 560

app.os.fileFindRoot

Finds the root base of file

```
<Function>
```

globals:

```
<Object> appe__config  
<Object> appe__control
```

arguments:

```
<String> filename  
<Boolean> inherit
```

returns:

```
<String>
```

position:

- [src/appe/src/js/os.js](#) line: 758

app.os.scriptOpen

Tries to open a script and load asynchronously, returns to callback

```
<Function>    // asynchronous
```

arguments:

```
<Function> callback  
<String> file  
<String> fn  
<Number> max_attempts
```

returns.

position:

- [src/appe/src/js/os.js](#) line: 806

app.os.generateFileHead

Generates a JSON head

```
<Function>
```

globals:

```
<Object> appe__config
```

arguments:

```
<Object> source  
<Date> timestamp
```

returns:

```
<Object>
```

position:

- [src/appe/src/js/os.js](#) line: 865

app.os.generateJsonChecksum

Generates a JSON checksum, returns to callback

```
<Function>    // asynchronous
```

arguments:

```
<Function> callback  
<String> source
```

returns.

position:

- [src/appe/src/js/os.js](#) line: 897

app.os.getLastFileName

Gets the file name of last opened file

```
<Function>
```

globals:

```
<Object> appe__config
```

returns:

```
<String>
```

position:

- [src/appe/src/js/os.js](#) line: 929

app.os.getLastFileVersion

Gets the runtime version of last opened file

```
<Function>
```

globals:

```
<Object> appe__store
```

returns:

```
<String>
```

position:

- [src/appe/src/js/os.js](#) line: 957

app.os.getLastFileChecksum

Gets the JSON checksum of last opened file

<Function>

globals:

<Object> appe__store

returns:

<String>

position:

- [src/appe/src/js/os.js line: 990](#)

app.os.getLastFileHead

Gets the last opened file header

<Function>

returns:

<Object>

position:

- [src/appe/src/js/os.js line: 1022](#)

app.controller

app.controller

Controller functions

```
<Object>
```

app.controller.spoof

Captures the app position using location.href

```
<Function>
```

arguments:

```
<Object> loc { view, action, index }
```

position:

- [src/appe/src/js/controller.js](#) line: 16

app.controller.history

Handles history navigation through the browser History API

```
<Function>
```

arguments:

```
<String> title  
<String> url
```

position:

- [src/appe/src/js/controller.js](#) line: 54

app.controller.cursor

Get or set the controller cursor,

it contains the current position in the app

```
<Function>
```

arguments:

```
<Object> loc
```

returns:

```
<Object> loc { view, action, index }
```

position:

- [src/appe/src/js/controller.js](#) line: 99

app.controller.setTitle

Set and store the document title

```
<Function>
```

arguments:

```
<String> title
```

returns:

```
<String>
```

position:

- [src/appe/src/js/controller.js](#) line: 124

app.controller.getTitle

Gets the document title

```
<Function>
```

returns:

```
<String>
```

position:

- [src/appe/src/js/controller.js](#) line: 140

app.controller.store

Stores data from current session, returns to callback

```
<Function> // asynchronous
```

globals:

```
<Object> appe__store
```

arguments:

```
<Function> callback  
<String> fn  
<Object> schema  
<Object> data
```

returns.

position:

- [src/appe/src/js/controller.js line: 157](#)

app.controller.retrieve

Restores data for current session and loads file and scripts, returns to callback

```
<Function> // asynchronous
```

globals:

```
<Object> appe__store
```

arguments:

```
<Function> callback  
<Array> routine
```

returns.

position:

- [src/appe/src/js/controller.js line: 228](#)

app.controller.clear

Reset the current session data

```
<Function>
```

globals:

```
<Object> appe__config  
<Object> appe__store
```

returns:

```
<Boolean>
```

position:

- [src/appe/src/js/controller.js](#) line: 306

app.memory

app.memory

Handles storage entries

`<Object>`

available methods:

```
- set (key <String>, value)
- get (key <String>)
- has (has <String>, value)
- del (key <String>)
- reset ()
```

app.memory.set

Sets storage entry

`<Function>`

arguments:

```
<String> key
value
```

returns.

position:

- [src/appe/src/js/memory.js line: 25](#)

app.memory.get

Gets storage entry

`<Function>`

arguments:

```
<String> key
value
```

returns.

position:

- [src/appe/src/js/memory.js](#) line: 39

app.memory.has

Checks existence for storage entry by key and match value

<Function>

arguments:

<String> key
value

returns:

<Boolean>

position:

- [src/appe/src/js/memory.js](#) line: 53

app.memory.del

Removes storage entry by key

<Function>

arguments:

<String> key

returns.

position:

- [src/appe/src/js/memory.js](#) line: 66

app.memory.reset

Resets storage

<Function>

returns.

position:

- [src/appe/src/js/memory.js](#) line: 78

app.store

app.store

Handles persistent storage entries

`<Object>`

available methods:

```
- set (key <String>, value)
- get (key <String>)
- has (has <String>, value)
- del (key <String>)
- reset ()
```

app.store.set

Sets persistent storage entry

`<Function>`

arguments:

```
<String> key
value
```

returns.

position:

- [src/appe/src/js/store.js line: 25](#)

app.store.get

Gets persistent storage entry by key

`<Function>`

arguments:

```
<String> key
```

returns.

position:

- [src/appe/src/js/store.js line: 38](#)

app.store.has

Checks existence for persistent storage entry by key and match value

<Function>

arguments:

<String> key
value

returns:

<Boolean>

position:

- [src/appe/src/js/store.js line: 52](#)

app.store.del

Removes persistent storage entry by key

<Function>

arguments:

<String> key

returns.

position:

- [src/appe/src/js/store.js line: 65](#)

app.store.reset

Resets persistent storage

<Function>

returns.

position:

- [src/appe/src/js/store.js line: 77](#)

app.start

app.start

Launcher functions

<Object>

app.start.redirect

Tries to redirect after a delay

<Function>

arguments:

<Boolean> loaded

position:

- [src/appe/src/js/start.js line: 16](#)

app.start.alternative

Displays message with info and alternatives to help to execute the app

<Function>

globals:

<Object> appe__config

returns.

position:

- [src/appe/src/js/start.js line: 37](#)

app.start.load

Default "start" load function

<Function>

globals:

```
<Object> appe__config  
<Object> appe__locale
```

returns.

position:

- [src/appe/src/js/start.js](#) line: 96

app.start.progress

Displays the current loader status

```
<Function>
```

arguments:

```
<Number> phase
```

position:

- [src/appe/src/js/start.js](#) line: 227

app.start.loadComplete

Fires on "start" load complete

```
<Function>
```

globals:

```
<Object> appe__config
```

arguments:

```
<Object> routine
```

returns.

position:

- [src/appe/src/js/start.js](#) line: 256

app.main

app.main

"main" functions

```
<Object>
```

app.main.control

Init "main" function that fires when "main" is ready

```
<Function>
```

globals:

```
<Object>  appe__config
```

arguments:

```
<Object>  loc
```

returns.

position:

- [src/appe/src/js/main.js line: 18](#)

app.main.handle

Control "main" function handling requests, could return self prototype

```
<Function>  prototype    constructor
```

available prototype methods:

```
- getID ()
- setAction ()
- getAction ()
- setTitle (title)
- getTitle ()
- setMsg (msg)
- getMsg ()
- setURL (path, qs)
- redirect ()
- refresh ()
- resize ()
- selection ()
- export ()
- prepare ()
- prevent ()
- fetch ()
- open () <=> prepare ()
- add () <=> prepare ()
- edit () <=> prepare ()
- update () <=> prepare ()
- delete () <=> prevent ()
- close () <=> prevent ()
- history (reset)
- sender ()
- receiver ()
```

globals:

```
<Object> appe__config
<Object> appe__main
```

arguments:

```
<Event> e
```

returns.

position:

- [src/appe/src/js/main.js line: 156](#)

app.main.action

Actions "main", returns self prototype

```
<Function> prototype    constructor
```

available prototype methods:

```
- isInitialized (funcName)
- begin ()
- end ()
- menu (element)
```

globals:

```
<Object> appe__config
<Object> appe__main
```

arguments:

```
<Array> events  
<String> event  
<ElementNode> element
```

returns:

```
<Function> prototype
```

position:

- [src/appe/src/js/main.js line: 641](#)

app.main.load

Default "main" load function

```
<Function>
```

globals:

```
<Object> appe__config  
<Object> appe__locale
```

returns.

position:

- [src/appe/src/js/main.js line: 749](#)

app.main.beforeunload

Default "main" before unload function

```
<Function>
```

returns:

```
<Boolean>
```

position:

- [src/appe/src/js/main.js line: 869](#)

app.main.loadComplete

Fires on "main" load complete

```
<Function>
```

globals:

```
<Object>  appe__config
```

arguments:

```
<Object>  routine
```

returns.

position:

- [src/appe/src/js/main.js line: 889](#)

app.main.setup

Setup "main" data

```
<Function>
```

globals:

```
<Object>  appe__main
```

position:

- [src/appe/src/js/main.js line: 950](#)

app.view

app.view

"view" functions

```
<Object>
```

app.view.spoof

Captures the current position inside "view" using location.href

```
<Function>
```

returns:

```
<Object> loc { action, index }
```

position:

- [src/appe/src/js/view.js](#) line: 16

app.view.control

Control "view" function, returns self prototype

```
<Function> prototype constructor
```

available prototype methods:

```
- isInitialized (funcName)
- begin ()
- end ()
- setID (id)
- getID ()
- getLastID ()
- setEvent (event)
- getEvent ()
- setTitle (section_title, view_title, id)
- setActionHandler (label, id)
- denySubmit ()
- fillTable (table, data, order)
- fillForm (form, data)
- fillSelection (data, id)
- fillCTA (id)
- paginate (element, pages, current_page)
- localize (element)
```

globals:

```
<Object> appe__config
<Object> appe__control
<Object> appe__locale
```

arguments:

```
<Array> events
<Object> data
<ElementNode> form
```

returns:

```
<Function> prototype
```

position:

- [src/appe/src/js/view.js](#) line: 75

app.view.action

Actions "view", returns self prototype

```
<Function> prototype    constructor
```

available prototype methods:

```
- isInitialized (funcName)
- begin ()
- end ()
- getID ()
- validateForm ()
- prepare (data, submit)
- prevent (data, submit, title, name)
- open (data, submit) <=> prepare ()
- add (data, submit) <=> prepare ()
- edit (data, submit) <=> prepare ()
- update (data, submit)<=> prepare ()
- delete (data, submit, title, name) <=> prevent ()
- close (data, submit, title, name) <=> prevent ()
- selection ()
- print ()
```

globals:

```
<Object> appe__config
<Object> appe__control
<Object> appe__locale
```

arguments:

```
<Array> events
<String> event
<ElementNode> element
<ElementNode> form
```

returns:

<Function> prototype

position:

- [src/appe/src/js/view.js line: 645](#)

app.view.sub

Sub-actions "view", returns requested prototype method

<Function> prototype constructor

available prototype methods:

```
- csv (element, table)
- clipboard (element, table)
- toggler (element, dropdown)
```

globals:

<Object> appe__config

arguments:

```
<String> method
<ElementNode> element
<ElementNode> table
```

returns:

<Function>

position:

- [src/appe/src/js/view.js line: 991](#)

app.view.handle

Fires when "view" is loaded

<Function>

globals:

```
<Object> appe__config
<Object> appe__control
```

returns.

position:

- [src/appe/src/js/view.js line: 1151](#)

app.view.send

Sends control messages to "main"

```
<Function>
```

globals:

```
<Object> appe__config  
<Object> appe__control
```

arguments:

```
<Object> ctl
```

returns.

position:

- [src/appe/src/js/view.js line: 1188](#)

app.view.fetch

Fetch data from "main" store

```
<Function>
```

globals:

```
<Object> appe__control
```

arguments:

```
<String> from
```

returns.

position:

- [src/appe/src/js/view.js line: 1246](#)

app.view.resize

Fires when "view" is resized

```
<Function>
```

globals:

```
<Object>  appe__control
```

returns.

position:

- [src/appe/src/js/view.js](#) line: 1267

app.view.getFormData

Helper to get form data with transformation and sanitization

```
<Function>
```

arguments:

```
<HTMLCollection>  elements
```

returns:

```
<Object>
```

position:

- [src/appe/src/js/view.js](#) line: 1297

app.view.convertTableCSV

Helper to convert object data to csv text format

```
<Function>
```

arguments:

```
<ElementNode>  table
```

returns:

```
<String>
```

position:

- [src/appe/src/js/view.js](#) line: 1359

app.view.copyToClipboard

Helper to copy into system clipboard

links: <https://gist.github.com/rproenca/64781c6a1329b48a455b645d361a9aa3>

```
<Function>
```

arguments:

```
<String> source
```

returns.

position:

- [src/appe/src/js/view.js](#) line: 1412

app.view.load

Default "view" load function

```
<Function>
```

globals:

```
<Object> appe__config  
<Object> appe__locale  
<Object> appe__store
```

returns.

position:

- [src/appe/src/js/view.js](#) line: 1454

app.view.beforeunload

Default "view" before unload function

```
<Function>
```

globals:

```
<Object> appe__control
```

returns:

```
<Boolean>
```

position:

- [src/appe/src/js/view.js](#) line: 1680

app.view.loadComplete

Fires on "view" load complete

```
<Function>
```

globals:

```
<Object>  appe__config
```

arguments:

```
<Object>  routine
```

returns.

position:

- [src/appe/src/js/view.js](#) line: 1714

app.layout

app.layout

Handles layout functions

```
<Object>
```

app.layout.renderElement

Renders a document element

```
<Function>
```

arguments:

```
<String> node  
<String> content  
<Object> attributes
```

returns:

```
<String>
```

position:

- [src/appe/src/js/layout.js line: 19](#)

app.layout.renderSelect

Renders a SELECT element

```
<Function>
```

arguments:

```
<String> select_id  
<Object> data  
<Object> attributes
```

returns:

```
<String>
```

position:

- [src/appe/src/js/layout.js](#) line: 58

app.layout.renderSelectOption

Renders the SELECT element OPTION

```
<Function>
```

arguments:

```
<String> value  
<String> name  
<Boolean> selected
```

returns:

```
<String>
```

position:

- [src/appe/src/js/layout.js](#) line: 84

app.layout.renderSelectOptionGroup

Renders the SELECT element OPTGROUP

```
<Function>
```

arguments:

```
<String> label  
<String> options
```

returns:

```
<String>
```

position:

- [src/appe/src/js/layout.js](#) line: 102

app.layout.renderSelectOptions

Renders SELECT elements

```
<Function>
```

example:

```
[ { "optgroup_label": [ { "option_name": "option_value" }, ... ] } ]  
[ { "option_name": "option_value" }, ... ]  
[ "option_value", ... ]
```

arguments:

```
<String> select_id  
<Object> data
```

returns:

```
<String>
```

position:

- [src/appe/src/js/layout.js](#) line: 126

app.layout.dropdown

Helper for dropdown, returns requested prototype method

```
<Function> prototype    constructor
```

available prototype methods:

- open (e)
- close (e)
- toggle (e)

arguments:

```
<String> event  
<ElementNode> toggler  
<ElementNode> dropdown  
<Function> callback (e, dropdown)
```

returns:

```
<Function>
```

position:

- [src/appe/src/js/layout.js](#) line: 177

app.layout.collapse

Helper for collapsible, returns requested prototype method

```
<Function> prototype    constructor
```

available prototype methods:


```
- open (e)
- close (e)
- toggle (e)
```

arguments:

```
<String> event
<ElementNode> element
<ElementNode> collapsible
<Function> callback (e, collapsible)
```

returns:

```
<Function>
```

position:

- [src/appe/src/js/layout.js](#) line: 318

app.layout.draggable

Helper for draggable, returns requested prototype method

TODO: *FIX droid*

```
<Function> prototype    constructor
```

available prototype methods:

```
- start (e, row, callback)
- over (e, row, callback)
- enter (e, row, callback)
- leave (e, row, callback)
- end (e, row, callback)
- drop (e, row, callback)
```

arguments:

```
<String> event
<ElementNode> row
<String> row_selector - .draggable
<Function> callback (e, row)
```

returns:

```
<Function>
```

position:

- [src/appe/src/js/layout.js](#) line: 459

app.layout.localize

Helper to localize layout

```
<Function>
```

globals:

```
<Object>  appe__locale
```

arguments:

```
<ElementNode>  element
```

returns.

position:

- [src/appe/src/js/layout.js](#) line: 654

app.utils

app.utils

Utils functions

<Object>

app.utils.system

Detects system environment

<Function>

arguments:

<String> purpose (name | platform | architecture | release)

returns.

position:

- [src/appe/src/js/utils.js](#) line: 17

app.utils.addEvent

Helper to add element event listener

<Function>

arguments:

<String> event
<ElementNode> element
<Function> func

returns.

position:

- [src/appe/src/js/utils.js](#) line: 141

app.utils.removeEvent

Helper to remove element event listener

```
<Function>
```

arguments:

```
<String> event  
<ElementNode> element  
<Function> func
```

returns.

position:

- [src/appe/src/js/utlis.js line: 164](#)

app.utlis.proxy

Proxy function with passed arguments

```
<Function>
```

arguments:

```
<Boolean> deep  
<Object> | <Function> obj
```

returns:

```
<Object> |
```

position:

- [src/appe/src/js/utlis.js line: 186](#)

app.utlis.storage

Storage utility, it stores persistent (across the session) and non-persistent data

```
<Function> prototype    constructor
```

available prototype methods:

```
- set (key, value)  
- get (key)  
- has (key, value)  
- del (key)  
- reset ()  
- fake ()
```

arguments:

```
<String> persists
<String> method
<String> key
value
```

returns.

position:

- [src/appe/src/js/utlis.js](#) line: 232

app.utils.cookie

Helper to handle cookie

```
<Function> prototype    constructor
```

available prototype methods:

```
- set (key, value, expire_time)
- get (key)
- has (key, value)
- del (key)
- reset ()
```

arguments:

```
<String> method
<String> key
value
<Date> expire_time
```

returns.

position:

- [src/appe/src/js/utlis.js](#) line: 414

app.utils.base64

Base64 encoder and decoder

```
<Function> prototype    constructor
```

available prototype methods:

```
- encode (to_encode)
- decode (to_decode)
```

arguments:

```
to_encode
```

returns.

position:

- [src/appe/src/js/utils.js](#) line: 568

app.utils.transform

Transforms type of passed value

```
<Function>
```

arguments:

```
<String> purpose  ( lowercase | uppercase | numeric | integer | json )  
value
```

returns.

position:

- [src/appe/src/js/utils.js](#) line: 630

app.utils.sanitize

Sanitizes passed value

```
<Function>
```

arguments:

```
<String> purpose  ( whitespace | breakline | date | datetime | datetime-local | array )  
value
```

returns.

position:

- [src/appe/src/js/utils.js](#) line: 659

app.utils.classify

Transforms object to classnames

```
<Function>
```

arguments:

```
<Object> | <Array> data  
<String> prefix  
<Boolean> to_array
```

returns:

```
classes
```

position:

- [src/appe/src/js/utils.js](#) line: 711

app.utils.numberFormat

Formats number float within decimal and thousand groups

```
<Function>
```

arguments:

```
<Number> number  
<Number> decimals  
<String> decimals_separator  
<String> thousands_separator
```

returns:

```
<String>
```

position:

- [src/appe/src/js/utils.js](#) line: 750

app.utils.dateFormat

Formats date, supported format specifiers are like used in strftime() C library function,

it accepts Date time format or boolean true for 'now', default: "Y-m-d H:M"

```
<Function>
```

format specifiers:

```
- d // Day of the month, digits preceded by zero (01-31)
- J // Day of the month (1-31)
- w // Day of the week (1 Mon - 7 Sun)
- m // Month, digits preceded by zero (01-12)
- n // Month (1-12)
- N // Month, start from zero (0-11)
- Y // Year, four digits (1970)
- y // Year, two digits (70)
- H // Hours, digits preceded by zero (00-23)
- G // Hours (0-23)
- M // Minutes, digits preceded by zero (00-59)
- I // Minutes (0-59)
- S // Seconds, digits preceded by zero (00-59)
- K // Seconds (0-59)
- v // Milliseconds, three digits
- a // Abbreviated day of the week name (Thu)
- b // Abbreviated month name (Jan)
- x // Date representation (1970/01/01)
- X // Time representation (01:00:00)
- s // Seconds since the Unix Epoch
- V // Milliseconds since the Unix Epoch
- O // Difference to Greenwich time GMT in hours (+0100)
- z // Time zone offset (+0100 (CEST))
- C // Date and time representation (Thu, 01 Jan 1970 00:00:00 GMT)
- Q // ISO 8601 date representation (1970-01-01T00:00:00.000Z)
```

arguments:

```
<Date> | <Boolean> time
```

returns:

```
formatted_date
```

position:

- [src/appe/src/js/utlis.js](#) line: 804

app.utlis.numberLendingZero

Pads number from left with zero

```
<Function>
```

arguments:

```
number
```

returns.

position:

- [src/appe/src/js/utlis.js](#) line: 866

app.utlis.isPlainObject

Checks if object is a plain object

(jQuery.fn.isPlainObject)

jQuery JavaScript Library

links: <https://jquery.com/>

license: MIT license <https://jquery.org/license>

copyright: Copyright JS Foundation and other contributors

```
<Function>
```

arguments:

```
<Object> obj
```

returns:

```
<Boolean>
```

position:

- [src/appe/src/js/utils.js](#) line: 886

app.utils.extendObject

Deep extend and merge objects

(jQuery.fn.extend)

jQuery JavaScript Library

links: <https://jquery.com/>

license: MIT license <https://jquery.org/license>

copyright: Copyright JS Foundation and other contributors

```
<Function>
```

returns:

```
<Object> target
```

position:

- [src/appe/src/js/utils.js](#) line: 923

Hooks

start.session

<Function>

arguments:

<Function> callback

position:

- [src/appe/src/js/index.js line: 396](#)

main.session

<Function>

arguments:

<Function> callback

position:

- [src/appe/src/js/index.js line: 404](#)

control.session

<Function>

arguments:

<Function> callback

position:

- [src/appe/src/js/index.js line: 412](#)

start.loadComplete

<Function>

arguments:

<Object> routine

position:

- [src/appe/src/js/start.js](#) line: 165

start.alternative

<Function>

position:

- [src/appe/src/js/start.js](#) line: 181

main.handle

<Function>

arguments:

```
<Function> prototype  
<String> event  
<Object> ctl
```

position:

- [src/appe/src/js/main.js](#) line: 215

main.setup

<Function>

arguments:

```
<Object> data
```

position:

- [src/appe/src/js/main.js](#) line: 958

control.renderRow

<Function>

arguments:

```
<ElementNode> throw_tpl  
<Number> id  
<Object> data[id]  
<Object> args
```

position:

- [src/appe/src/js/view.js line: 432](#)

control.fillForm

<Function>

arguments:

<Object> data
<Object> args

position:

- [src/appe/src/js/view.js line: 473](#)

control.openView

<Function>

arguments:

<Object> data

position:

- [src/appe/src/js/view.js line: 1169](#)

control.loadComplete

<Function>

arguments:

<Object> routine

position:

- [src/appe/src/js/view.js line: 1493](#)