

O.Backoffice

Version 1.0

What is backoffice?

- In short, admin page.

Layers, why not?

- ▮ Important layers:
 - ▮ **admin**: administration pages, each has main and detail pages.
 - ▮ **js/listener**: listeners of events of DOMs.
 - ▮ **js/backoffice**: services at back office.
 - ▮ **js/api**: APIs to web services
- ▮ Other layers:
 - ▮ **css**: all styles
 - ▮ **img**: all images
 - ▮ **js**: all javascript libraries
 - ▮ **js/common**: common functions
 - ▮ **js/util**: utility functions
- ▮ Start page:
 - ▮ home.html

Hierarchy

- BaseListener
 - Direct Known Subclasses:
 - HomeListener
 - BookListener
 - AuthorListener
 - TagListener
- BaseBackoffice
 - Direct Known Subclasses:
 - HomeBackoffice
 - BookBackoffice
 - AuthorBackoffice
 - TagBackoffice
- BaseAPI
 - Direct Known Subclasses:
 - BookAPI
 - AuthorAPI
 - TagAPI

Composition

- xxxAPI
- Common
- Converter

Module in set and in order

1. Admin Page

- Main page
- Detail page

2. Listener

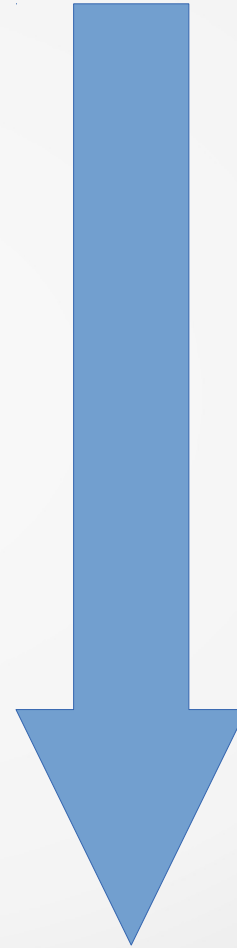
- Corresponding listener

3. Backoffice

- Corresponding backoffice

4. API

- Corresponding api



Main Page

- `class="storeData"`
- DOM has data to store

Detail Page

- `class="requestData"`
- DOM has data to send

xxxBackoffice @override

- addNewRecordEvent
- viewLnkRecordEvent
- editLnkRecordEvent
- deleteLnkRecordEvent
- editBtnRecordEvent
- deleteBtnRecordEvent
- cancelBtnRecordEvent
- saveBtnRecordEvent
- applyFilterRecordsEvent
- clearFilterRecordsEvent
- filterBtnRecordsEvent

- recordAPI
- initRecords
- initRecord
- getTemplateValue
- getTemplateDom
- preInitRecords
- preInitRecord
- postInitRecords
- postInitRecord
- initFilters
- setMandatoryFields
- validateRecord
- getRequestData
- giveDomData
- reloadPage
- previousPagerRecord
- nextPagerRecord

Start Point

- home.html
- once it loads all scripts :
 - jQuery
 - Bootstrap
 - Common
 - Converter
 - API
 - Backoffice
 - Listener
 - etc.

Corresponding Pages

- Loaded DOM
- Main page calls initRecords
- Detail page calls initRecord

Demo