Contents

[Task 1. Setup 1](#_Toc391474060)

[Task 2. Get Bookmarks from the server 1](#_Toc391474061)

[Task 3. Create Bookmark 2](#_Toc391474062)

[Task 4. Edit Bookmark 2](#_Toc391474063)

[Task 5. Delete Bookmark 3](#_Toc391474064)

[Task 6. Get tags from the server 3](#_Toc391474065)

[Task 7. Filter bookmarks by Tag 3](#_Toc391474066)

# Task 1. Setup

* Create folder “backbone\_basics” in the root directory of your web server and unzip the task.zip to that folder.
* The structure of our SPA webServerRoot/backbone\_basics/

**additional/app.jpg** – screen short of our SPA

**additional/backbone\_basic.sql** – sql for creating database and input some initial data

**api/\*** - api php scripts for models and collections

**css/bootstrap/\*** – bootstrap framework for styling our SPA

**js/app.js** – our SPA (here we will write code)

**lib/\***- libraries for our SPA: jquery.js, underscore.js, backbone.js

**php-classes/\*** - php classes for access to MySQL DataBase

**index.html** – start page.

* Create var app.

# Task 2. Get Bookmarks from the server

* Setup the following params
  + Backbone.emulateHTTP = true;
  + Backbone.emulateJSON = true;
* Create Models:
  + app.models
  + app.models.BookmarkModel
    - defaults
    - urlRoot
    - initialize
  + app.models.TagModel
* Create Collections:
  + app.collections
  + app.collections.BookmarkListCollection
    - model
    - url
  + app.collections.TagListCollection
    - model
    - url
* Create Views:
  + app.views
  + app.views.BookmarkView
    - tagName,
    - template
    - render
  + app.views.BookmarkListView
    - initialize (only listenTo “sync”, subviews)
    - render
    - show
* Create Router:
  + app.routers
  + app.routers.Router
    - initialize
    - routes (default)
    - default
    - \_loadBookmarkListCollection
* Create Objects:
  + app.collections.bookmarkListCollection
  + app.views.bookmarkListView
  + app.routers.router

# Task 3. Create Bookmark

* Modify Model:
  + app.models.BookmarkModel
    - validate
* Create app.GlobalEvent
* Create View:
  + app.views.BookmarkFormView
    - el
    - initialize
    - events
    - render
    - renderAlert
    - saveBookmark
    - clearForm
* Modify View:
  + app.views.BookmarkListView
    - app.GlobalEvent.on(“saveBookmark”)
    - refresh
* Create Objects:
  + app.models.bookmarkEditModel
  + app.views.bookmarkFormView

# Task 4. Edit Bookmark

* Modify View:
  + app.views.BookmarkView
    - events (Edit)
    - editBookmark

# Task 5. Delete Bookmark

* Modify Views:
  + app.views.BookmarkView
    - events (Delete)
    - deleteBookmark
  + app.views.BookmarkListView
    - app.GlobalEvent.on(“deleteBookmark”)

# Task 6. Get tags from the server

* Create Model:
  + app.models.TagCountModel
* Create Collection:
  + app.collections.TagCountListCollection
    - model
    - url
* Create Views:
  + app.views.TagCountView
    - tagName
    - template
    - render
  + app.views.TagCountListView
    - el
    - initialize
    - render
    - refresh
* Modify Router
  + app.routers.Router (add \_loadTagCountListCollection)
    - modify default
    - \_loadTagCountListCollection
* Create Objects:
  + app.collections.tagCountListCollection
  + app.views.tagCountListView

# Task 7. Filter bookmarks by Tag

* Create View:
  + app.views.BookmarkListContainerView
    - el
    - events
    - clearFilter
* Modify Router:
  + app.routers.Router
    - routes (add “filter/:tag router”)
    - bookmarkFilter
* Create Object:
  + app.views.bookmarkListContainerView