



Educational Offering

JavaScript User Interface Development Using Backbone.js

Course Requirements: JavaScript programming knowledge

Environment Requirements: Windows/Mac/Unix/Linux Laptop or a Desktop with internet access

Topic Outline

Supporting Frameworks

This module will describe the role that frameworks play in supporting single page application (SPA) user interface development, including:

- DOM Manipulation/Ajax Support - jQuery
- Dependency Management Modularity - Require.js
- MVC and Navigation - Backbone.js with Marionette
- HTML Templating and Form Binding - `_underscore/handlebars`, form binding with `Stikit`
- Mocking API Layer - `MockJax`

Application Structure / Deployment Model

This module will describe approaches for deploying JavaScript and HTML resources using one of the following HTTP server environments:

- JEE App Server
- Apache
- IIS
- Node.js
- Nginx
- Jetty

Folder structure for library, JavaScript, and HTML resources will also be discussed

Exercise #1 - Create application project file/folder structure under static HTTP server, as well as define static server.

Loading Dependent Libraries and Application Startup

This module will discuss how the sole index.html is loaded, and starts the initial bootstrapping of the JavaScript application.

Exercise #2 - Implement index.html along with Require.js library path configuration, application object, and startup logic.

Navigation/Router/Controllers

This module will describe Backbone.js's routing-based navigation. This feature allows for bookmarkable URLs that are deep linked into application user interface views. The Marionette framework provides additional support by introducing controllers that provide additional event handling behavior.

Exercise #3 - Define and implement Backbone.js router using a Marionette controller.

Collections and Models

This module will describe how Backbone models and collections are used to model application data, provide asynchronous CRUD support, and provide synchronization capabilities with server side data models using RESTful APIs. The following will be discussed:

- Defining a Model
- Fetching, Posting, Put, and Delete (CRUD)
- Defining a Collection of Models
- Adding and Removing
- Synchronizing with Server Side
- Defining and Handling Events, Detecting Changes
- Getting/Setting Properties, Transforming To and From JSON
- Iterating Over Collections (Underscore Support)

Exercise #4 - MockJax will be used to “mock” server side JSON/API so the application models/collections can be defined and exercised to reinforce topics discussed in this section.

Backbone/Marionette Views

The Marionette framework generalizes common Backbone View (UI) compositions. Each of the view types will be discussed and applied along with the lifecycle, initialization, and rendering of template-based HTML user interfaces. Topics covered:

- Initializing
- Event Handling
- Rendering HTML Templates (Underscore/Handlebars)
- Item Views
- Binding Models to Forms, Validating and Saving (Stickit)

Exercise #5 - Defining a basic item view to display.

- Collection Views
- Composite Views

Exercise #6 - Applying Collection and Composite Views.

Note: All exercises described in this outline build upon each other. At the completion of this course, students will have a working reference application.