

Web Essentials: Backbone

Course Summary

Description

Web Essentials: Backbone covers Backbone and provides practical and hands-on experience with the preferred JavaScript library for building rich web applications. If you are building for the web, learning Backbone will change the way you write JavaScript, working across multiple browsers and platforms.

Objectives

At the end of this course, students will be able to:

- Learn what Backbone is and how to add it to your applications
- Use Backbone to select complex sets of elements from the DOM
- Develop rich web pages that respond to user interaction
- Interact with your server-side code using Ajax
- Explore a wide variety of plugins and learn how to write your own

Topics

- Introduction to Backbone
- Views
- Templates
- Models
- Collections
- Routers
- Additional Topics in Backbone

Audience

This is an introductory and beyond-level web development course, designed for experienced JavaScript / web developers who need to extend their knowledge of web design and development.

Prerequisites

This is a hands-on programming class. Attendees should have previous experience or working knowledge of developing software applications. Real world programming experience is a must.

Duration

Two days

Web Essentials: Backbone

Course Outline

I. Introduction to Backbone

- A. Definition of MVC
- B. Views and routers as controllers in Backbone
- C. Why use Backbone.js
- D. Comparison to other JavaScript libraries

II. Views

- A. Using multiple views
- B. Creating custom view classes
- C. Initializing the view
- D. Removing a view
- E. Optional: modal views

III. Templates

- A. Underscore.js templating language
- B. Setting up the template with script tags
- C. Underscore.js delimiters
- D. Using JavaScript in an Underscore.js template
- E. Other templates: Mustache and Handlebars

IV. Models

- A. Extending Backbone.Model
- B. Initializing a model
- C. Getters and setters
- D. Model events
- E. Maintaining model state
- F. RESTful persistence

V. Collections

- A. Backbone.Collection
- B. Initializing a collection
- C. Store and retrieve a collection
- D. Filtering collections
- E. Useful collection methods

VI. Routers

- A. Backbone.Router
- B. Initializing a route
- C. Processing the route
- D. Backbone.history.start()
- E. Using Backbone.sync to maintain route state

VII. Additional Topics in Backbone

- A. Creating modular applications with RequireJS
- B. Asynchronous Module Definition (AMD)
- C. Traditional unit testing with QUnit
- D. Behavior-Driven Development testing with Jasmine
- E. Managing a large code base with Marionette
- F. Managing a large code base with Thorax