

Core Backbone.js

Course Summary

Description

Backbone.js was the first of the major JavaScript MVC (or MV*) frameworks to achieve huge adoption with startups and established enterprises alike. Today, Backbone (along with Ember and Angular) remains one of the most used frameworks for building flexible, powerful, well-organized JavaScript applications.

Backbone distinguishes itself through a focus on providing straightforward components that a developer can combine in the way that is most valuable to him or her. Backbone involves less "magic" than other popular frameworks and can easily work together with existing code. A robust ecosystem means developers can add more capabilities through plug-ins - either authoring their own, or using popular open-source add-ons.

Objectives

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

- Understand how client-side JavaScript MVC applications are designed and implemented
- Learn the infrastructure that Backbone relies upon
- Learn the elements and patterns which Backbone provides for building MVC applications
- Look at options and best practices for organizing code and getting the most out of Backbone
- Learn about integrating server data, deployment, and testing

Topics

- Review of OO / Advanced JavaScript (optional)
- Review of jQuery (optional)
- Introduction to Client-Side MVC Concepts
- Introducing Backbone.js
- Working with the Server
- Modular Development
- Extensions and Helpers
- Best Practices and Tools
- Testing Approaches with Backbone.js

Audience

This course is designed for web developers.

Prerequisites

To take this course you should have:

- Solid knowledge of HTML, JavaScript and jQuery
- Some familiarity with CSS
- Some familiarity with the MVC design pattern (from client, server, desktop, or mobile apps)

Duration

Three days

Core Backbone.js

Course Outline

- I. **Review of OO / Advanced JavaScript (optional)**
- II. **Review of jQuery (optional)**
 - A. DOM, AJAX, Events, Promises/Deferreds
- III. **Introduction To Client-Side MVC**
- IV. **Concepts**
 - A. Models, Views, Controllers, Templates, Events
- V. **Introducing Backbone.js**
 - A. Infrastructure
 1. Underscore, inheritance, mixins
 - B. Views and DOM events
 - C. Templates
 - D. Models
 - E. Collections
 - F. Event-based designs
 - G. Routing: Control-flow and history
- VI. **Working with the server**
- VII. **Modular development**
- VIII. **Extensions and helpers**
- IX. **Best practices and tools**
- X. **Testing approaches with Backbone.js**