

Ember.js Essentials

Course Summary

Description

Ember.js calls itself “a framework for creating ambitious web applications.” In this course, we teach developers how to create a single-page web application using model-view-controller-router architecture. We will use Handlebars.js as our templating library.

Objectives:

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

- Understand the benefits of a single-page web application
- Use the model-view-controller-router pattern
- Write advanced JavaScript syntax for the Ember library and its dependencies
- Use Handlebars.js to dynamically insert data into the DOM

Topics

- Introduction
- Ember vs. other JavaScript libraries
- The Ember Application Object
- The Object Model
- Templates
- Link Helper
- Action Helper
- Input Helper
- Routing
- Controllers
- Models
- Views
- Components
- Enumerables
- Testing

Audience

Ember.js training is for experienced web developers who have a solid understanding of HTML, JavaScript and jQuery, and some exposure to CSS. Familiarity with the concept of an MVC pattern is helpful, but covered in class.

Prerequisites

Attendees should have a solid working knowledge of advanced JavaScript, as well as HTML, CSS, and jQuery. Familiarity with using JavaScript for server side communication with AJAX is good but not necessary.

Software Needed on Each Student PC

- A recent version of Google Chrome or Mozilla Firefox or other modern browser is best
- Text editor or IDE with JavaScript syntax support, such as Notepad++
- Related lab files that ProTech will provide

Duration

Two days

Ember.js Essentials

Course Outline

I. Introduction

- A. General JavaScript syntax for Ember
- B. Objects
- C. Properties
- D. Methods
- E. Array literals
- F. Object literals
- G. Overview of MVC: Model, View, Controller
- H. Downloading Ember.js
- I. Who is Ember.js for?
- J. Naming conventions

II. Ember vs. other JavaScript libraries

III. The Ember Application Object

IV. The Object Model

- A. Classes and Instances
- B. Computed Properties
- C. Aggregate Data
- D. Observers
- E. Bindings
- F. Mixins

V. Templates

- A. The Application Template
- B. The Handlebars.js Templating Language
- C. Variables
- D. Conditionals
- E. Iterating over a List
- F. Binding Element Attributes
- G. Binding Element Class Names
- H. Helpers

VI. Link Helper

VII. Action Helper

- A. Registering DOM Event Listeners

VIII. Input Helper

- A. Bound vs. Unbound Templates

IX. Routing

- A. Creating the Application's Router
- B. Mapping URLs to Routes
- C. Query Strings
- D. Configuring the Router
- E. Generated Routes
- F. Specifying a Route's Model
- G. Configuring a Route's Controller
- H. Rendering the Route's Template
- I. Redirecting
- J. Asynchronous Routing

X. Controllers

- A. Defining Controllers
- B. Providing Controllers with Models
- C. Rendering Dynamic Data from Controllers
- D. ObjectController Single Model
- E. ArrayController Multiple Models
- F. Handling Event Actions
- G. Dependencies Between Controllers
- H. State Transitions in Controllers

XI. Models

- A. Defining a Model
- B. Create and Delete Records
- C. Storing Records
- D. Persisting Records
- E. Finding Records
- F. REST Adapter for Data Persistence

XII. Views

- A. Define a View
- B. Register Event Handlers in Views
- C. Put Views into Templates
- D. Add Layouts to Views
- E. Customize a View's Element Tag
- F. Built-in Views

XIII. Components

- A. Defining a Component
- B. Component Properties
- C. Component Content
- D. Customizing a Component's Element Tag
- E. Using Components to Send Actions

XIV.Enumerables

XV. Testing

- A. Unit Testing