Ember

NSCC | W0400010

Prog2700

Levis,Gabriel

2024

Table of Contents

[What is Ember.js? 2](#_Toc164134781)

[Key Features of Ember.js: 2](#_Toc164134782)

[Convention over Configuration: 2](#_Toc164134783)

[Ember CLI: 2](#_Toc164134784)

[Routes: 2](#_Toc164134785)

[Components: 2](#_Toc164134786)

[Handlebars Template System: 3](#_Toc164134787)

[Explaining how routes are used to define different parts of an application and how they work with templates to render content. 3](#_Toc164134788)

[Commands and steps for Workshop: 4](#_Toc164134789)

[Conclusion 5](#_Toc164134790)

[Links 6](#_Toc164134791)

# What is Ember.js?

Ember.js is an open-source JavaScript framework designed for developing sophisticated, feature-rich web applications in a maintainable and productive way. It follows the Model-View-ViewModel (MVVM) pattern and is known for its robustness, scalability, and convention-over-configuration approach. Ember is built with the idea that web developers can create large, complex applications that are easy to manage and evolve over time.

## Key Features of Ember.js:

### Convention over Configuration:

Ember.js prefers convention over configuration, meaning it provides a standard way of doing things, which reduces the amount of decision-making and configuration necessary from developers. This leads to faster development and less room for errors as developers follow a unified structure and practice.

### Ember CLI:

Ember Command Line Interface (CLI) is a powerful tool that helps with everything from creating new projects and developing apps to building and serving your application. It's an essential part of the Ember ecosystem that streamlines development processes, enforces best practices, and provides a suite of useful tools and utilities (like code generators, build tools, and dependency management).

### Routes:

Routing in Ember.js is a core concept that manages the application state and URLs. It is responsible for loading data, handling URL changes, and rendering templates. Routes define the logic that runs when certain URLs are accessed, ensuring that users have a bookmarkable, shareable URL that reflects the state of the application.

### Components:

Components in Ember are reusable and isolated elements that manage their own functionality and layout. They help in building a complex user interface that is composed of small, manageable parts, each encapsulating their own behavior and design. Components in Ember can be easily integrated with data management solutions, supporting complex application features.

### Handlebars Template System:

Ember uses Handlebars, a semantic templating language, to efficiently describe the UI of an application in a declarative way. Handlebars templates are written with a mixture of HTML and Handlebars expressions, which make it easy to dynamically update the HTML displayed to the user in response to data changes. This template system is particularly powerful when combined with Ember's data binding features, allowing developers to build interactive applications effortlessly.

Together, these features make Ember.js a formidable choice for developers aiming to build large-scale, single-page web applications. Ember's architecture and ecosystem are designed to handle the complexities of modern web app development while providing a developer-friendly interface that boosts productivity and maintains high standards of quality.

## Explaining how routes are used to define different parts of an application and how they work with templates to render content.

**1. Defining Application Structure:**

**URL Mapping:** Routes in Ember.js are used to map specific URLs to corresponding application states. Each route corresponds to a particular URL or route segment. For example, a route named posts might correspond to the /posts URL, while a route named post might correspond to /posts/:post\_id, where :post\_id is a dynamic segment representing a specific post.

**Nested Routes:** Routes can be nested within each other to represent hierarchical application structures. For instance, you might have a blog route with nested posts and post routes to represent a typical blogging application structure.

**2. Loading Data:**

Model Hook: Routes in Ember.js often include a model hook, which is responsible for fetching data required for the corresponding template. Inside the model hook, you can make asynchronous requests to fetch data from APIs, databases, or other sources. Once the data is retrieved, it is made available to the template for rendering.

**3. Rendering Templates:**

Template Rendering: After the model hook (if present) has resolved and data is available, the route transitions to the associated template. Ember.js uses the Handlebars template system to define these templates. Templates are HTML files with embedded Handlebars expressions that allow dynamic content rendering based on the data provided by the route's model.

**Template Context:** The template is rendered within the context of the route, meaning it has access to the data loaded by the route's model hook. This allows for seamless integration of data into the template, enabling dynamic rendering of content based on the application state.

**Rendering Hierarchy:** Templates in Ember.js can be organized hierarchically to match the nested structure of routes. For example, if you have nested routes like blog.posts.post, you can define corresponding nested templates to render each level of the application's UI.

**Summary:**

In summary, routes in Ember.js define the structure and behavior of different parts of an application by mapping URLs to application states, loading data required for rendering templates, and coordinating the rendering of templates with dynamic content. By leveraging routes and templates, developers can create organized, maintainable, and interactive single-page web applications that respond dynamically to user actions and URL changes.

## Commands and steps for Workshop:

node --version

npm --version

<https://nodejs.org/en>

npm install -g ember-cli

ember -v

cd prog2700

mkdir ember

cd ember

ember new Hello-World

cd hello-world

ember generate route hello

Navigate to the generated route template at app/templates/hello.hbs and update it to display a "Hello World" message

<h1>Hello World</h1>

ember serve

## Conclusion

This short workshop is designed to give a hands-on introduction to Ember.js, focusing on the practical aspects of creating an application and understanding its structure and core concepts. The workshop should ignite participants' interest in Ember and provide them with the confidence to explore more complex features and applications on their own.

## Links

<https://nodejs.org/en>

<https://guides.emberjs.com/release/>

<https://github.com/emberjs/ember.js>

<https://selleo.com/blog/what-is-ember-js>

<https://www.tutorialspoint.com/emberjs/emberjs_overview.htm>

<https://medium.com/aeturnuminc/what-is-ember-js-ff94403fec96>

*Edited by: Marie-Josee Dassylva (Mother)*