

By OnlineInterviewQuestions.com

Ember js Interview questions

Here are a few Ember js Questions provided with their answers, which you may encounter while being interviewed and will help you crack the interview on Ember Js

Read Best Ember.js Interview Questions and Answers for Freshers and Experienced.

Q1. What is Ember.js? Write the steps to create an app in Ember.js?

Ember.js is an open source technology written in javascript based on a model view pattern and was developed by ember core team on 8th December 2011. These are the steps to create an application in ember.js is :

1. Firstly install an ember-cli. Almost all applications are built with ember-cli.
2. Create a new application by using ember new and an application will be generated.
3. Use materialize-CSS for styling to give a material design.
4. Create components by using ember g component.
5. Check whether there is a router.js file in which all of your routes are defined.
6. If I have a video route and I would like to display a set of youtube videos on the page then I am going to create a simple video card component that I will be iterating over and display on the video page.

Q2. Explain directory structure in Ember.js?

The new command generates a project structure also called directory structure with the following files and directories:

I-app:- This is where folders and files for models, components, routes, templates, and styles are stored.

I-bower_components/ bower.json:- Bower is a dependency management tool which is used in Ember CLI to manage front-end plugins and component dependencies.

I-config:- The config directory contains the environment.js where we can configure settings for your app.

I-dist:-When we build our app for deployment, the output files will be created here.

I-node_modules/package.json:- Directory and files are from npm. Npm is the package manager for node.js.

Public:- This directory contains assets such as image and fonts.

Vendor:- This directory is where front-end dependencies that are not managed by Bower go.

Tests/testem.js:- Automated tests for our app go in the test folder, and testem is configured in testem.js.

Tmp:- Ember CLI temporary files live here.

Ember-cli-build.js:- This file describes how Ember CLI should build our app.

Q3. Explain what is ember-data?

Ember comes with a data management library called Ember data which deals with application data that defines the structure of the data. We can generate ember-data model using Ember CLI. The main purpose of an ember-data is that it is a library that integrates tightly with ember.js to make it easy to retrieve records from a server, cache them for performance, save updates back to the server and create new records on the client.

Q4. What is ember route? How can you generate a route in ember.js?

An ember route is built with three parts:

1. An entry in the Ember router which maps between our route name and a specific URI.
2. A route handler file, which sets up what should happen when that route is loaded.
3. A route template, which is where we display the actual content for the page.

In ember when we want to make a new page that can be visited using a URL, we need to generate a “route” using Ember CLI. Hence the generator will print out:

1. Installing route
2. Create app/routes/about.hbs
3. Create app/templates/about.hbs
4. Updating router
5. Add route about
6. Installing route test

To define a route, run ember generate route route-name. This command will generate a file name route-name.js in app/routes/ folder.

Q5. Write command to start and stop development server in Ember.js?

We can install Ember with a single command using npm such as: `-npm install -g ember-cli@2.17`. We can use `ember new` command to create a new application:- `ember new ember-quickstart`. This command will create a new directory called `ember-quickstart` and set up a new Ember application inside it. Outside, the application will include a development server. We can start a development server by typing the command:

1. `Cd ember-quickstart`
2. `Ember serve`

To stop the development server at any time simply type `Ctrl-c` in our terminal.

Q6. What are observers in Ember.js?

Ember supports observing any property which also includes computed properties. Observers are something which contains the behavior that reacts to the changes made in other properties. Observers are used when we need to perform some behavior after a binding has finished synchronizing. New ember developers often use observers. Observers are mostly used within the ember framework and for that; computed properties are the appropriate solution. An observer can be set on an object using the following syntax- “`ember.observer`” Observers in ember are synchronous. They will fire as soon as they observe a change in of the properties. And, because of this, it is easy to introduce bugs where properties are not yet synchronized.

Q7. What is the component in ember.js? How is it defined?

A component is something which encapsulates certain snippets of handlebars templates that are to be used again and again. In such case, a JavaScript is not necessary to be written. Just define the entire handlebars template and use the component that is created. Components make it easy to reuse the code, create widgets, tags in or not in the W3C, and much more. Components in ember.js are basically versions of web components.

Defining a component in Ember.js: `Ember generate component component_name;`

Q8. What are enumerable in ember.js?

An enumerable is any object in ember.js which contains a number of child objects and allows us to work with those children using “`ember.enumerable`” API. The native JavaScript array is the most common enumerable in a majority of applications, which ember.js extends to conform to the enumerable interface. Ember.js provides us with a standardized interface for dealing with the enumerable and allows us to completely change the way our underlying data is stored without modifying the other parts of the application which accesses it. This API follows ECMAScript specifications as much as possible which hence minimizes the incompatibility with other libraries. It also allows ember.js to use the native browser implementations in arrays where they are available.

Q9. Explain model with reference to ember.js. Also, tell how it is defined in it.

Models are objects that represent the underlying data which is presented to the user by your application. Different applications have different objects or models, and it depends completely on what problem they are trying to solve. For example, a photo album is a collection of many photos i.e. group of photos. And, a photo sharing application has a photo model to represent a particular photo. Models are persistent and most of them are loaded from and saved to a server that uses the database to store data. Once the models are loaded from the storage, the components translate the data into the user interface with which the user can interact. Defining model in ember.js: –

Syntax-

```
ember generate model model_name;
```

Q10. What are services in ember.js?

It is a long-lived Ember object that can be made available in different parts of your application. It is created using the following syntax- “ember.service”

Example uses of **Ember.js services** include:

- Logging
- User/session authentication
- Geolocation
- Third-party API's
- Web Sockets
- Server sent events or notifications
- Server-backed API calls that may not fit ember-data.

Services are generated with the help of ember CLI's service generator. If you want to access a service, inject it either in an initializer or use the following syntax- “ember.inject”. Using this you can even access the properties and methods using services. Use the following syntax to define a service:

Syntax-

```
ember generate service service_name;
```

Q11. Which template library is used by Ember.js ?

Q12. Write steps to write a Handlebars Helper?

Q13. How to disable Prototype Extensions in Ember.js ?

Please Visit OnlineInterviewquestions.com to download more pdfs