By OnlineInterviewQuestions.com

Marionette js Interview Questions & Answers

Read Top Marionette js Interview Questions

Q1. What is Marionette.js ?

Marionette.js is a composite application library for the Backbone.js. It simplifies the construction of the large-scale JavaScript applications. It is nothing but a collection of the common design and implementation patterns found in applications. Marionette brings the architecture of an application to the Backbone. It also has built-in view management and memory management. The Marionette is a lightweight and flexible library tool. It sits on top of the backbone and provides the framework for building a scalable application.

Q2. List some features of Marionette.js ?

Features of Marionette.js

- It is scalable as the applications built-in modules with event-driven architecture.
- It is easily modifiable. It works with the specific need of your application.
- It has built-in memory management and zombie killing for View, CollectionViews and Region.
- It is an event-driven architecture. It is flexible.
- It can create application visuals at runtime with Region and View objects. It reduces boilerplate for all views.

Q3. What is current stable version of Marionette.js ?

v4.1.2 is the current stable version of MarionetteJS.

Q4. How to install and configure Marionette.js ?

It is installed using the npm package manager. Type the following command.

```
npm install backbone.marjonette
```

Marionette has a global configuration setting. It changes how the system works. **Marionette.VERSION** and **Marionette.DEV_MODE** are some of the configuration properties available in Marionette.

Q5. How can you intialize Marionette.js ?

The **Backbone.Marionette.Application** object is used to initialize among other things the various pieces of your application. The Initialize is called immediately after the application has been instantiated. It is invoked with the same arguments that the constructor received.

Example

```
var MyApp = Marionette.Application.extend({
  initialize: function(options) {
    console.log(options.container);  } });
var myApp = new MyApp({container: '#app'});
```

Q6. Explain how to trigger and listen Events in Marionette.js ?

The events can be triggered using **Marionette.trigger** method. It can also be used to do additional processing of your application. There are two events that are triggered. They are, Before start - It is fired before your application is started and before the initializer is executed. Start - It is fired after the application starts and after the initializer is executed.

Example

```
MyApp.on("before:start", function(options){
  options.moreData = "Event firing in Marionette"
});
MyApp.on("start", function(options){
  if (Backbone.history){
    Backbone.history.start();
  }
});
```

Q7. How to configure Routes in Marionette.js ?

The routes are configured in the **appRoutes**. The definition of the route is passed to Backbone's standard routing handlers. You have to provide a callback method that exists on the controller instead of a callback function that exists on the router.

Example of configuration routes with appRoute

```
 var MyRouter = Backbone.Marionette.AppRouter.extend({
  // "someMethod" must exist at controller.someMethod
```

```
  appRoutes: {
    "some/route": "someMethod"
  },
  /* standard routes are mixed with appRoutes/Controllers above */
  routes : {
    "some/otherRoute" : "someOtherMethod"
  },
});
```

The routes can also be configured in the constructor.

Example

```
var MyRouter = new Marionette.AppRouter({
  controller: myController,
  appRoutes: {
    "foo": "doFoo",
    "bar/:id": "doBar"
  }
});
```

Q8. How to set a Renderer in Marionette.js?

Renderer in Marionette is used to render data into a template. It can be set by the View class by using the class method setRendered. It accepts two arguments. The first is the template that is passed to the view. The second is the data that is to be rendered into the template. This function returns a string that contains the result of applying data to the template.

Example

```
Marionette.View.setRenderer(function(template, data) {
  return _.template(template)(data);
});
var myView = new Marionette.View({
  template: 'Hello <%- name %>!',
  model: new Backbone.Model({ name: 'World' })
});
myView.render();
myView.el === '<div>Hello World!</div>';
```

Q9. What Marionette getTemplate function does ?

The **get template function in the Marionette** is used to choose a template to render. It renders after the view has been instantiated. It can be used to change the template based on simple logic like the value of a specific attribute in the view model. The return value of the function can either be a jQuery selector or a compiled template function.

Example

```
var Mn = require('backbone.marionette');
var MyView = Mn.View.extend({
  getTemplate: function(){
    if (this.model.get('is_active')){
      return '#template-when-active';
    } else {
      return '#template-when-inactive';
    }
  }
});
```

Q10. How to create a Model in Marionette?

The model can be rendered using the backbone Marionette and can be attached to our views. After attaching, it can be used to render the data that they represent.

Example

```
var Bb = require('backbone');
var MyModel = Bb.Model.extend({
  defaults: {
    name: 'world' }
});
```