

[By OnlineInterviewQuestions.com](http://OnlineInterviewQuestions.com)

Backbone js Interview Questions

Top Backbone js interview questions and Answers for freshers and experienced.

Below are the list of top 18 **Backbone js interview questions**. Here You can read interview questions on Backbone js.

Q1. Explain what is Backbonejs?

Backbone.js is a client-side (Model, View, Controller) MVC-based [JavaScript framework](#). It purely is written in Javascript.

Backbone.js gives structure to web applications by providing models with key-value binding and custom events, collections with a rich API of enumerable functions views with declarative event handling, and connects it all to your existing API over a RESTful JSON interface.

Q2. List Dependencies of using BackboneJs?

Backbone's Js only hard dependency is [Underscore.js](#) ($\geq 1.8.3$). For RESTful persistence and DOM manipulation with Backbone. View, include [jQuery](#) ($\geq 1.11.0$), and json2.js for older Internet Explorer support. (Mimics of the Underscore and jQuery APIs, such as Lodash and Zepto, will also tend to work, with varying degrees of compatibility.)

Q3. What is the use of BackboneJS route?

The Backbone.js router is used to change the URL fragment or hashes of an application to create bookmarkable and shareable URLs.

Q4. What is a collection in Backbone.js?

In Backbone.js, the collection is used to represent ordered set of models. Any event in model triggers an event in collection directly. For example, you can bind “change” event to be notified in a case when any model in the collection has been modified.

Q5. What are the main BackboneJs native components?

The main native components of BackboneJs are:-

- Model
- View
- Collection
- Router
- Event class object

Q6. Explain Events in BackboneJs?

Backbone events is a module that can be mixed into any object, giving the object the ability to bind and trigger custom named events. Events are not declared before they are bound to any object. Events reflect the state of the model.

Q7. What is the difference between the properties “id” and “cid” on a model object in Backbone Js?

The “**id**” property on a model is automatically assigned based on the “id” set in the model’s attributes hash.

Ideally, this is the ID that you receive from the rest API for the resource that you are querying. On the other hand, “**cid**” is an ID temporarily assigned to each model and is useful until an actual ID is determined for the object. For example, a model pushed to a collection that has not yet been persisted can be addressed using “cid”, until it is saved in the database and an actual ID is generated for it.

Q8. List out configuration options available in Backbone Js?

- InitialCopyDirection
- modelSetOptions
- change Triggers
- boundAttribute
- suppressThrows
- converter

Q9. What is Modelbinder in Backbone Js ?

ModelBinder class is used to make synchronization process of views and models together.

Q10. What is a converter in Backbone Js ?

When a model's attribute is copied to an HTML element or when an HTML element's value is copied into a model's attribute, a function is called, this function is known as a converter in Backbone Js.

Also, Read **Best 25 [React js Interview questions](#)**

Q11. Explain Architecture of Backbone?

Below image shows the Architecture of Backbone.js

Architecture of Backbone

Image not found or type unknown

Q12. What is Backbone.sync is used for?

Backbone.sync is a function that is Backbone js call every time it attempts to read or save a model to the server. By default, it uses **jQuery.ajax** to make a RESTful JSON request and returns a **jqXHR**

Example Code

```
Backbone.sync = function(method, model) {
  alert(method + ": " + JSON.stringify(model));
  model.set('id', 1);
};
var author= new Backbone.Model({
  author: "Sharad",
  website: "https://www.onlineinterviewquestions.com"
});
author.save();
author.save({author: "Teddy"});
```

Q13. Explain what is ModelBinder in Backbone.js?

ModelBinder is a class in Backbone.js to bind backbone model attributes to:

- Read-only html elements such as , <div> etc.
- Html element attributes such as enabled, displayed, style etc.
- Editable form elements such as <input>, <textarea> etc. This type of binding is bidirectional between the HTML elements and the Model's attributes.

Q14. How to access a models data from a view in Backbone.js?

In order to access a models data from a view in backbone.js you have to initialize your Model :

```
var author= Backbone.Model.extend({
  initialize: function(){
    console.log('intailized');
  },
  defaults:{
    names:['Bob', 'Sim', 'Dart']
  }
});
var person_view = Backbone.View.extend({
  initialize: function() {
    this.model = new author();
  },
  output: function(){
    console.log(this.model.get('names'))
  }
});
```

Q15. How are models attributes stored in Backbone.js?

In Backbone.js models attributes stored in a **hash**.

Q16. What is Router in Backbone? How do you create a Router with Backbone?

Backbone Router is used for routing client-side applications and connects them to actions and events using URLs. Backbone.Router is used to create a Router with Backbone.

Q17. What is Collection in Backbone?

A collection is a set of models which binds events when the model has been modified in the collection. The collection contains a list of models that can be processed in the loop and supports sorting and filtering. When creating a collection, we can define what type of model that collection is going to have along with the instance of properties. Any event triggered on a model, which will also trigger on the collection in the model.

Q18. How to Create a Model In Backbone js?

Creating a Model in Backbone Js

```
Person = Backbone.Model.extend({
  initialize: function(){
    alert("Welcome to Backbone Js");
  }
});
var person = new Person;
```

Please Visit OnlineInterviewquestions.com to download more pdfs