

➤ (<https://www.educba.com/ruby-interview-questions-and-answer/>)

◀ (<https://www.educba.com/red-hat-interview-questions/>)



Introduction to Redux Interview Questions and Answers

React is a User interface library. The core Redux is built upon action, state, and reducers. All the data resides at the store. Reacts components use actions in order to update the stores. reducers help the store to update itself. Props make sure the updated stores are available for the React.

Now Redux is a knowable state container built for **JavaScript apps** (<https://www.educba.com/java-and-javascript/>). it allows managing application state and it stands to be open source. applications which run consistently help, run in different environments (client, server, and native), and are easy to test.

^

Now, if you are looking for a job which is related to Redux then you need to prepare for the Redux Interview Questions. It is true that every interview is different as per the different

job profiles but still to clear the interview you need to have a good and clear knowledge of Redux. Here, we have prepared the important Redux Interview Questions And Answers which will help you get success in your interview.

Below are the 10 important Redux Interview Questions And Answers that are frequently asked in an interview. these questions are divided into two parts are as follows:

Part 1 – Redux Interview Questions (Basic)

This first part covers basic Redux Interview Questions and Answers

1) Benefits of Redux?

**EDUCBA ALL IN ONE
BUNDLE - Lifetime PLAN**

₹24999
~~₹249999~~

Get Unlimited Access to all 5511 Courses
& 204 Job Oriented Programs

Offer Expires in:

00 D **23** H

26 M **32** S

LEARN MORE

**([HTTPS://WWW.EDUCBA.COM/FULL-
ACCESS-BUNDLE](https://www.educba.com/full-access-bundle))**

Answer:

Maintainability: maintenance of Redux becomes easier due to strict code structure and organisation.

Organization: code organisation is very strict hence the stability of the code is high which intern increases the work to be much easier.

Server rendering: This is useful, particularly to the preliminary render, which keeps up a better user experience or **search**

engine optimization (<https://www.educba.com/search-engine-optimization-seo-basics/>). The server-side created stores are forwarded to the client side.

Developer tools: It is Highly traceable so changes in position and changes in the application all such instances make the developers have a real-time experience.

Ease of testing: The first rule of writing testable code is to write small functions that do only one thing and that are independent. Redux's code is made of functions that used to be: small, pure and isolated.

Q2) How Distinct from MVC and Flux?

Answer:

As far as MVC structure is concerned the data, presentation and logical layers are well separated and handled. here change to an application even at a smaller position may involve a lot of changes through the application. this happens because data flow exists bidirectional as far as MVC is concerned. Maintenance of MVC structures are hardly complex and Debugging also expects a lot of experience for it.

Flux stands closely related to redux. A story based strategy allows capturing the changes applied to the application state, the event subscription, and the current state are connected by means of components. Call back payloads are broadcasted by means of Redux.

Q3) Functional programming concepts?

Answer:

The various functional programming concepts used to structure Redux are listed below,

^

- Functions are treated as First class objects.
- Capable to pass functions in the format of arguments.

- Capable to control flow using, recursions, functions and arrays.
- helper functions such as reduce and map filter are used.
- allows linking functions together.
- The state doesn't change.
- Prioritize the order of executing the code is not really necessary.

Let us move to the next Redux Interview Questions.

Q4) Redux change of state?

Answer:

For a release of an action, a change in state to an application is applied, this ensures an intent to change the state will be achieved.

Example:

1. The user clicks a button in the application.
2. A function is called in the form of component
3. So now an action gets dispatched by the relative container.
4. This happens because the prop (which was just called in the container) is tied to an action dispatcher using `mapDispatchToProps` (in the container).
5. Reducer on capturing the action it intern executes a function and this function returns a new state with specific changes.
6. The state change is known by the container and modifies a specific prop in the component as a result of the `mapStateToProps` function.

Q5) Where can Redux be used?

Answer:

Redux is majorly used is a combination with reacting. it also has the ability to get used with other view libraries too. some of the famous entities like **AngularJS** (<https://www.educba.com/angularjs-interview-questions-for-experienced/>), **Vue.js** (<https://www.educba.com/vue-js-vs-jquery/>), and **Meteor**. can get combined with Redux easily. This is a key reason for the popularity of Redux in its ecosystem. So many articles, tutorials, middleware, tools, and boilerplates are available.

Part 2 – Redux Interview Questions (Advanced)

Let us now have a look at the advanced Redux Interview Questions.

Q6) What is the typical flow of data in a React + Redux app?

Answer:

Call-back from UI component dispatches an action with a payload, these dispatched actions are intercepted and received by the reducers. this interception will generate a new application state. from here the actions will be propagated down through a hierarchy of components from Redux store. The below diagram depicts the entity structure of a redux+react setup.

Q7) What is store in redux?

Answer:

The store holds the application state and supplies the helper methods for accessing the state.
register listeners and dispatch actions. There is only one Store while using Redux. The store is configured via the

createStorefunction. The single store represents the entire state. R

ducers return a state via action

```
export function configureStore(initialState) {  
  return createStore(rootReducer, initialState);  
}
```

The root reducer is a collection of all reducers in the application.

```
const rootReducer = combineReducers({  
  donors: donorReducer,  
});
```

Let us move to the next Redux Interview Questions.

Q8) Explain Reducers in Redux?

Answer:

The state of a store is updated by means of reducer functions. A stable collection of a reducers form a store and each of the stores maintains a separate state associated for itself. To update the array of donors, we should define donor application Reducer as follows.

```
export default function donorReducer(state = [], action) {  
  switch (action.type) {  
    case actionTypes.addDonor:  
      return [...state, action.donor];  
    default:  
      return state;  
  }  
}
```

^

The initial state and action are received by the reducers. Based on the action type, it returns a new state for the store.

The state maintained by reducers are immutable. The below-given reducer it holds the current state and action as an argument for it and then returns the next

```
state:function handlingAuthentication(st, actn)
{
return _.assign({}, st,
{
auth: actn.pyload
});
}
```

Q9) Redux workflow features?

Answer:

Reset: Allow to reset the state of the store

Revert: Roll back to the last committed state

Sweep: All disabled actions that you might have fired by mistake will be removed

Commit: makes the current state the initial state

Q10) Explain action's in Redux?

Answer:

Actions in Redux are functions which return an action object. The action type and the action data are packed in the action object. which also allows a donor to be added to the system. Actions send data between the store and application. All information's retrieved by the store are produced by the actions.

```
export function addDonorAction(donor) {
return {
type: actionTypes.addDonor,
donor,
};
}
```

Internal Actions are built on top of Javascript objects and associate a type property to it.

Recommended Article

This has been a guide to List Of Redux Interview Questions and Answers so that the candidate can crackdown these Redux Interview Questions easily. Here in this post, we have studied about top Redux Interview Questions which are often asked in interviews. You may also look at the following articles to learn more -

1. OS Interview Questions
(<https://www.educba.com/os-interview-questions/>)
2. VB.NET Interview Questions
(<https://www.educba.com/vb-net-interview-questions/>)
3. jQuery Interview Questions
(<https://www.educba.com/jquery-interview-questions/>)
4. Bootstrap Interview Questions
(<https://www.educba.com/bootstrap-interview-questions/>)
5. Bootstrap vs JQuery: What are the Differences
(<https://www.educba.com/bootstrap-vs-jquery/>)
6. Bootstrap vs WordPress and their benefits
(<https://www.educba.com/bootstrap-vs-wordpress/>)
7. Bootstrap vs jQuery UI: What are the Features
(<https://www.educba.com/bootstrap-vs-jquery-ui/>)
8. Bootstrap 4 Cheat sheet
(<https://www.educba.com/bootstrap-4-cheat->