Netcracker

Front-end development course

JavaScript

Lecture #10

State management

Application level state management - is a core problem in developing modern web applications. Many frameworks (like Angular or Ember) has state management model out of the box, but React is just a view library, so we should handle state management by ourselves.

We can invent our own architecture model for state management and implement it, or choose from existing libraries (or approaches), some of them are:

- Flux (pattern, designed by Facebook)
- Redux (implementation of Flux pattern)
- Mobx

Redux

Redux is state management library (written by Dan Abramov) implements (with slight differences) Flux architecture pattern and usually is used alongside with React to manage state and create dynamic applications.

Redux philosophy can be described in following sentences: the whole state of your app is stored in an object tree inside a single **store**. The only way to change the state tree is to emit an **action**, an object describing what happened. To specify how the actions transform the state tree, you write pure **reducers**.

Official documentation link.

Good article in Russian.

Single page application

SPA - is relatively new approach of implementing client-side applications. It involves implementing following things:

- Client-side routing
- Http backend for front-end
- Actual client-side application (with dynamic loading of views)
- Correct routing inside HTTP web server (Apache on nginx)
- Single HTML file (index.html)

React router

React router - is a third party library (that means it’s authors not directly connected with React) that implements client-side routing pattern and makes its usage available with React. React router has a pretty straightforward API and is easy to use.

Official guide link.

Good article in Russian to start from.

Q&A

Thank You

Netcracker