

AJAX with React

Ustun Ozgur

2015-05-26 Tue

Ajax with React

- ▶ React is agnostic wrt to Ajax
- ▶ It is just a view layer
- ▶ Just do the AJAX request, for example using jQuery's get function
- ▶ Register the response data in components state.
 - ▶ In the success handler, call setState.

componentDidMount

- ▶ This is a special lifecycle method like render.
- ▶ Called when the component is mounted to the real DOM.
- ▶ The technique: Initialize the state when component first mounts, then initiate an AJAX call when it is mounted.
- ▶ After the AJAX call, if the component is still mounted, update the state.

Example: Github

- ▶ Our aim is to retrieve and show the list of repos of a user.
- ▶ Github has an endpoint at `https://api.github.com/users/USER_NAME/repos` where `USER_NAME` is a placeholder.
- ▶ This returns a json response, an array of repos.

Empty Repos Array from getInitialState

```
var Repos = React.createClass({
  getInitialState: function () {
    return {repos: []};
  },
  render: function () {
    var renderRepo = function (repo) {
      return <div>Repo {repo.name} {repo.date}</div>
    }
    return <div>
      {this.state.repos.map(renderRepo)}
    </div>

  }

});
```

Retrieving data from Github

```
var GITHUB_ENDPOINT = "https://api.github.com/users/ustun/repositories";
var Repos = React.createClass({
  getInitialState: function () { return {repos: []};},

  componentDidMount: function () {
    $.getJSON(GITHUB_ENDPOINT, function (data) {
      this.setState({repos: data});
    }.bind(this));  },
  render: function () {
    var renderRepo = function (repo) {
      return <div>Repo {repo.name} {repo.created_at}</div>
    }
    return <div>{this.state.repos.map(renderRepo)}</div>
  }
});
```

Making sure that the component is still mounted

- ▶ When we initiate the request, the component is mounted,
- ▶ When the AJAX call returns, we are not sure.
- ▶ React might be trying to set state on a component that is no longer mounted.
- ▶ To guard against that, we use `isMounted` before setting state.

```
componentDidMount: function () {  
  $.getJSON(GITHUB_ENDPOINT, function (data) {  
    if (this.isMounted()) {  
      this.setState({repos: data});  
    }  
  }.bind(this)); },
```

Summary

- ▶ Simply call `setState` in response to AJAX responses.
- ▶ A usual place to initiate AJAX calls is in `componentDidMount`.
- ▶ AJAX calls can be initiated in event handlers too, it does not matter.

Exercise

- ▶ Instagram has an API at `https://api.instagram.com/v1/tags/nofilter/media/recent?client_id=CLIENT_ID` where `nofilter` is the name of the tag.
- ▶ For the workshop, you can use `CLIENT_ID=8ed8f14f61b24c95ad54583110ace715` or you can register your own Instagram app at `https://instagram.com/developer/`
- ▶ Implement an Instagram client that shows the photos tagged as HDR.
- ▶ The API returns a limited subset of responses, but has a `next` parameter. Add a "Load More" button at the bottom that loads more items from the API.
- ▶ Instagram implements same origin policy. To disable, -
`open -a Google\ Chrome --args --disable-web-security`
- ▶ `chrome.exe --disable-web-security`