**Московский государственный технический университет им. Н.Э.Баумана**

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Отчет по лабораторной работе №6 по курсу**

**РИП**

«React»

7

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы **ИУ5-52** |  |
|  |  |
| **Писарчук Н.** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |

Москва, МГТУ - 2018

Задание и порядок выполнения

В этой лабораторной работе вы засетапите React-приложение. Вам нужно будет создать одну страницу со списком репозиториев на гитхабе.

Для сдачи вы должны иметь:

1. Настроенный конфиг webpack и babel
2. React-приложение с одной страницей.
3. На странице должны отображаться репозитории с github. Репозитории должны загружаться с помощью axios через открытый api гитхаба.
4. Приложение должно успешно собираться вебпаком
5. Приложение должно содержать css
6. Приложение в dev режиме должно обновляться без перезагрузки страницы с помощью webpack-dev-server’а
7. index.html должен добавляться с помощью HtmlWebpackPlugin

Код

Repos.js

**import** React **from** 'react';  
**import** { hot } **from** 'react-hot-loader';  
**import** axios **from** 'axios';  
**import** Title **from** './Title/title';  
**import** ListItem **from** './LI/ListItem';  
  
**import** './Style.css';  
  
**export class** App **extends** React.Component {  
 constructor(props) {  
 **super**(props);  
 **this**.state = {  
 title: 'Мои репозитории',  
 color: 'black',  
 repos: [],  
 isLoading: **false**,  
 };  
  
 **this**.onChangeColor = **this**.onChangeColor.bind(**this**);  
 **this**.onRepoClick = **this**.onRepoClick.bind(**this**);  
 }  
 componentDidMount() {  
 **this**.setState({  
 isLoading: **true** });  
  
 axios.get('https://api.github.com/users/Nadi0998/repos')  
 .then(response => ({  
 repos: response.data,  
 isLoading: **false** }))  
 .catch(error => ({ isLoading: **false** }))  
 .then(data => **this**.setState(data));  
 }  
 onChangeColor() {  
 **this**.setState({  
 color: **this**.state.color === 'red' ? 'black' : 'red'  
 });  
 }  
 onRepoClick(repo) {  
 **this**.setState({  
 title: repo.name  
 })  
 }  
 render() {  
 **const** {  
 color,  
 title,  
 repos,  
 isLoading  
 } = **this**.state;  
  
 **return** (  
 <div>  
 <Title color={ color }>  
 { title }  
 </Title>  
  
 { isLoading && <h4>Загрузка...</h4> }  
  
 { repos.map((repo) => (  
 <ListItem  
 key={ repo.id }  
 repo={ repo }  
 onRepoClick={ **this**.onRepoClick }  
 />  
 )) }  
  
 <button onClick={ **this**.onChangeColor }>Изменить цвет</button>  
 </div>  
 )  
 }  
}  
**export default** hot(module)(App);

package.json

{  
 "name": "normal-lab6",  
 "version": "1.0.0",  
 "description": "",  
 "main": "src/index.js",  
 "scripts": {  
 "test": "echo \"Error: no test specified\" && exit 1",  
 "start": "webpack-dev-server --mode=development --open --hot",  
 "build": "webpack --mode=production"  
 },  
 "author": "",  
 "license": "ISC",  
 "dependencies": {},  
 "devDependencies": {  
 "@babel/core": "^7.2.2",  
 "@babel/preset-env": "^7.2.3",  
 "@babel/preset-react": "^7.0.0",  
 "axios": "^0.18.0",  
 "babel-cli": "^6.26.0",  
 "babel-loader": "^8.0.5",  
 "babel-preset-env": "^1.7.0",  
 "babel-preset-react": "^6.24.1",  
 "css-loader": "^2.1.0",  
 "html-webpack-plugin": "^3.2.0",  
 "prop-types": "^15.6.2",  
 "react": "^16.7.0",  
 "react-dom": "^16.7.0",  
 "react-hot-loader": "^4.6.3",  
 "style-loader": "^0.23.1",  
 "webpack": "^4.28.2",  
 "webpack-cli": "^3.1.2",  
 "webpack-dev-server": "^3.1.14"  
 }  
}

Index.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Lab6</title>  
</head>  
<body>  
 <div id="root"></div>  
 <script src="bundle.js"></script>  
</body>  
</html>

Index.js

import React from 'react';  
import ReactDOM from 'react-dom';  
import App from './Repos'  
ReactDOM.render(  
 <App/>,  
 document.getElementById('root')  
);

Style.css

button {  
 background: #ba2013;  
 color: white;  
 cursor: pointer;  
 padding: 10px;  
 border-radius: 3px;  
 margin-top: 10px;  
 outline: none;  
}  
div {  
 background: rgba(58, 255, 58, 0.24);  
}

Результат выполнения



