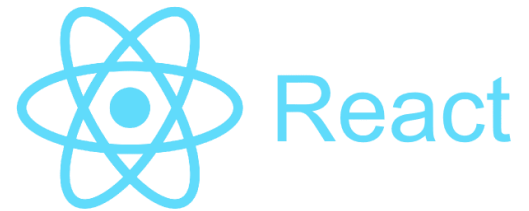




Microsoft Partner
Silver Learning



React Essential

Введение в роутинг



ITVVDN
IT VIDEO DEVELOPERS NETWORK

React Essential

Introduction

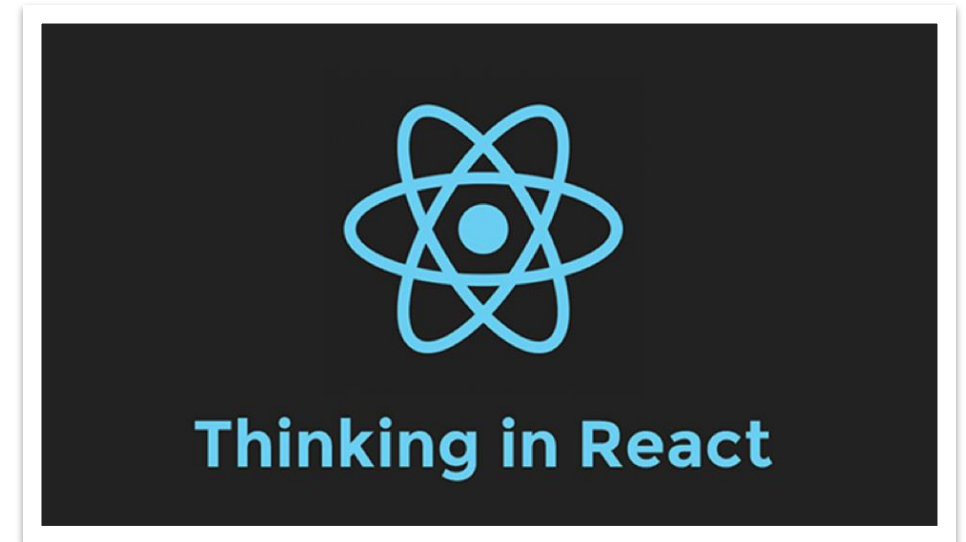


Муляк Дмитрий

Front-end developer at GlobalTechMakers

 dmitriymuliak

 dmitriymuliak



Введение в роутинг

Введение в роутинг

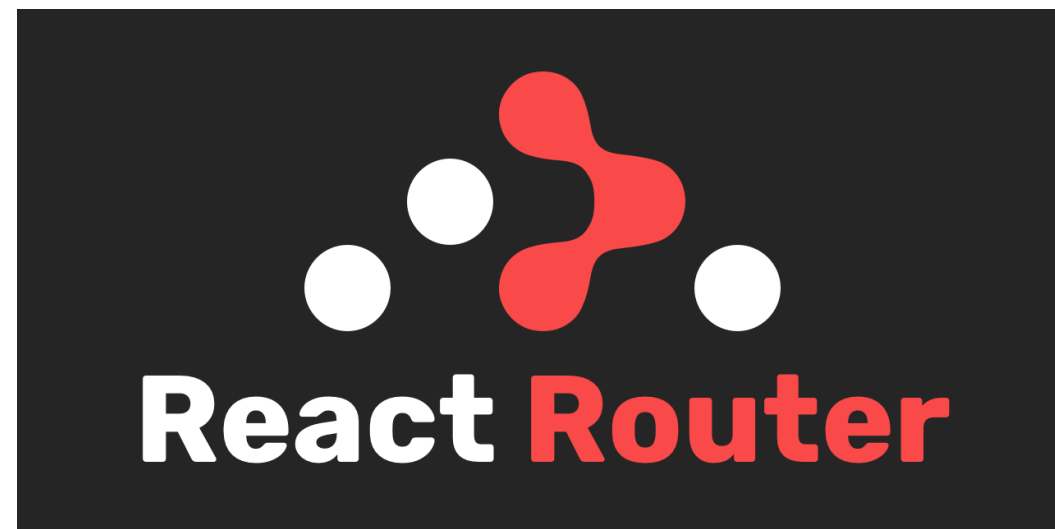
1. Обзор react-router
2. Создание страниц
3. Создание layouts

Обзор react-router

Маршрутизация

Пользователь должен видеть где он находится в данный момент времени в приложении, а также иметь возможность навигации по истории. Сам по себе React маршрутизацию не поддерживает, зато есть мощные библиотеки с целым арсеналом возможностей.

Эта библиотека популярна, довольно проста в использовании и обладает [хорошей документацией](#).



[react-router-github](https://github.com/reactjs/react-router)

[туториал React Router v4](#)

reacttraining.com

[React Router v5](#)

[туториал React Router v6](#)

React Essential

React Router Packages

Package	Version	Docs	Description
<code>react-router</code>	npm <code>v5.2.0</code>	API Docs <code>site</code> API Docs <code>markdown</code>	The core of React Router
<code>react-router-dom</code>	npm <code>v5.2.0</code>	API Docs <code>site</code> API Docs <code>markdown</code>	DOM bindings for React Router
<code>react-router-native</code>	npm <code>v5.2.0</code>	API Docs <code>site</code> API Docs <code>markdown</code>	React Native bindings for React Router
<code>react-router-config</code>	npm <code>v5.1.1</code>	API Docs <code>readme</code>	Static route config helpers

Redux users: The `react-router-redux` package is now deprecated. See [Redux Integration](#) for a better approach.

React Essential

Рендеринг Router

Компонент Router ожидает только один элемент в качестве дочернего. Чтобы работать в рамках этого условия, удобно создать компонент `<App/>`, который будет рендерить всё ваше приложение (это так же важно для серверного рендеринга).

```
import { BrowserRouter } from 'react-router-dom';

ReactDOM.render((
  <BrowserRouter>
    <App />
  </BrowserRouter>
), document.getElementById('root'))
```

React Essential

Routes

`<Route/>` компонент - это главный строительный блок React Router'a. В том случае, если вам нужно рендерить элемент в зависимости от pathname URL'ов, то следует использовать компонент `<Route/>`.

`<Route />` принимает path в виде prop, который описывает определенный путь и сопоставляется с location.pathname.

Когда текущий location.pathname сопоставляется положительно с prop path, то компонент будет отрендерен, а если мы не можем их сопоставить, то Route ничего не рендерит.

```
<Route path='/roster' />
// Когда location.pathname это '/', prop path не совпадает
// Когда location.pathname это '/roster' или '/roster/2', prop path со
// впадает
// Если установлен exact prop. Совпадает только строгое сравнение '/ro
// ster', но не
// '/roster/2'
<Route exact path='/roster' />
```



React Essential

Сопоставление пути

npm пакет `path-to-regexp` компилирует `prop path` в регулярное выражение и сопоставляет его против `location.pathname`. Строки `path` имеют более сложные опции форматирования чем объясняются здесь. Вы можете почитать [документацию](#).

Когда пути сопоставляются создается объект `match`, который содержит свойства:

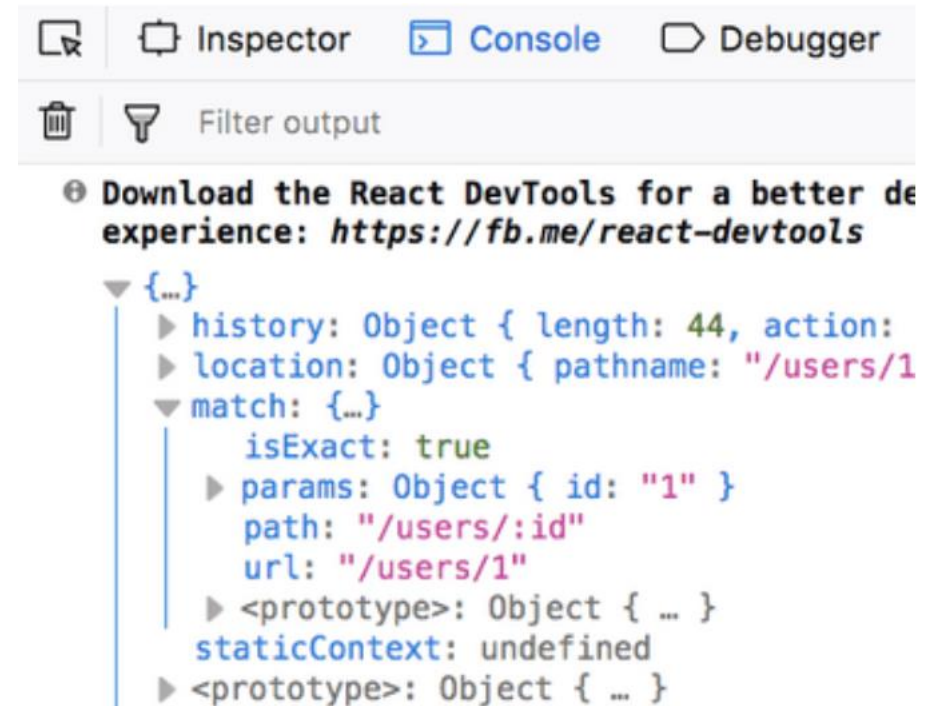
`url` — сопоставляемая часть текущего `location.pathname`;

`path` — путь в компоненте `Route`;

`isExact` — `path` в `Route` === `location.pathname`;

`params` — объект содержит значения из `path` которые возвращает модуль `path-to-regexp`.

- `path` в `Route` должен быть абсолютным.
- Можете поиграться с [тестером роутов](#) и посмотреть как создается объект `match`.



React Essential

Создание страниц

Компонент `Route` может быть в любом месте в роутере, но иногда нужно определять, что рендерить в одно и тоже место. В таком случае следует использовать компонент группирования `Route`'ов — `<Switch/>`.

`<Switch/>` итеративно проходит по дочерним компонентам и рендерит только первый, который подходит под `location.pathname`.

По порядку сопоставления путей в нашем приложении, все что нам нужно сделать - это создать компонент `Route` с `prop path`, который мы хотим сопоставить.

```
<Switch>
  <Route exact path="/" component={Home}/>
  {/* Оба /roster и /roster/:number начинаются с /roster */}
  <Route path="/roster" component={Roster}/>
  <Route path="/schedule" component={Schedule}/>
</Switch>
```



React Essential

Рендер компонента в Route

У Route есть 3 props'a, которые описывают каким образом выполнить рендер, сопоставляя prop path с location.pathname, и только один из prop должен быть представлен в Route:

component — React компонент. Когда роут удовлетворяет сопоставление в path, то он возвращает переданный component (используя функцию React.createElement).

render — функция, которая должна вернуть элемент React. Будет вызвана, когда удовлетворится сопоставление в path. Render довольно похож на component, но используется для inline рендеринга и подстановки необходимых для элемента props.

children — в отличие от предыдущих двух props children будет всегда отображаться независимо от того, сопоставляется ли path или нет.

```
<Route path='/page' component={Page} />
const extraProps = { color: 'red' }
<Route path='/page' render={(props) => (
  <Page {...props} data={extraProps}/>
)} />
<Route path='/page' children={(props) => (
  props.match
    ? <Page {...props}/>
    : <EmptyPage {...props}/>
)} />
```



React Essential

Создание layouts

React Router Config - помощник настройки статического маршрута для React Router. (не является обязательным к использованию вместе с библиотекой «react-router»).

Чтобы гарантировать сопоставление path за пределами render() с «matchRoutes» и внутри render(), вы должны использовать renderRoutes (внутри ваших компонентов) вместо <Route>.

```
ReactDOM.render(  
  <BrowserRouter>  
    { /* kick it all off with the root route */  
      {renderRoutes(routes)}  
    }  
  </BrowserRouter>,  
  document.getElementById("root")  
);
```

```
import { renderRoutes } from "react-router-config";  
  
const routes = [  
  {  
    component: Root,  
    routes: [  
      {  
        path: "/",  
        exact: true,  
        component: Home  
      },  
      {  
        path: "/child/:id",  
        component: Child,  
        routes: [  
          {  
            path: "/child/:id/grand-child",  
            component: GrandChild  
          }  
        ]  
      }  
    ]  
  }  
];
```

[React Router Config](#)



Проверка знаний

TestProvider.com



Проверьте как Вы усвоили данный материал на TestProvider.com

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

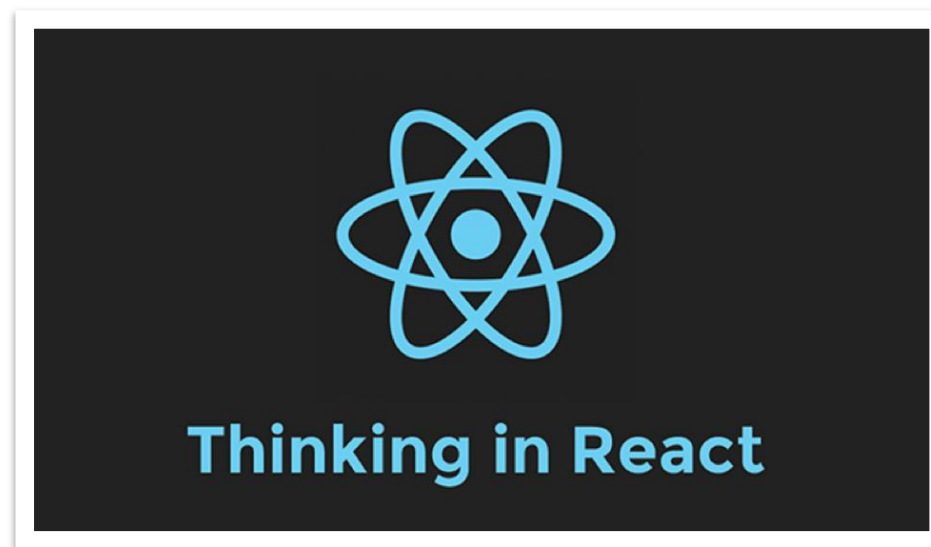
React Essential

Спасибо за внимание! До новых встреч!



Муляк Дмитрий

Front-end Developer at GTM



Информационный видеосервис для разработчиков программного обеспечения

