Введение в роутинг

№ урока: 8 **Курс:** React Essential

Средства обучения: Браузер Chrome, редактор кода VS Code или любой другой.

Обзор, цель и назначение урока

Познакомиться с библиотекой react-router. Научиться организовывать маршрутизацию в приложении.

Изучив материал данного занятия, учащийся сможет

- Создавать маршрутизацию в приложении.
- Создавать роуты вместе с <Route> и <Switch> компонентами.
- Создавать роуты вместе с библиотекой react-router-config.

Содержание урока

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

Резюме

- **Маршрутизация** - пользователь должен видеть то, 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.
- **Router** - при старте проекта вам нужно определить какой тип роутера использовать. Для браузерных проектов есть *BrowserRouter* и *HashRouter* компоненты.
- **BrowserRouter** — следует использовать, когда вы обрабатываете на сервере динамические запросы, а HashRouter используйте тогда, когда у вас статический веб сайт.
- **История** — History. Каждый Router создает объект *history,* который хранит путь к текущему location и перерисовывает интерфейс сайта, когда происходят какие-то изменения пути. Остальные функции предоставляемые в React Router полагаются на доступность объекта *history* через context, поэтому они должны рендериться внутри компонента Router. Компоненты React Router, не имеющие в качестве предка компонент Router, не будут работать, так как не будет доступен context.
- **Рендеринг Router**. Компонент Router (BrowserRouter) 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).
- **<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/>. Принимает prop ***exact***, благодаря которому пути сравниваются строго. Вы можете использовать компонент <Route/> без path.

CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- **Path** — <Route /> принимает path в виде prop, который описывает определенный путь и сопоставляется с location.pathname. Path может быть либо строкой, либо массивом из строк. Когда речь идет о пути React, Router думает только о пути без домена. Это значит, что в адресе:

  «http://www.example.com/my-projects/one?extra=false» - React Router будет видеть только «/my-projects/one».

- **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 те, что объясняются здесь. Вы можете ознакомиться с ними в документации.

- Когда пути сопоставляются создается объект ***match*** который содержит свойства:

  - **url** — сопоставляемая часть текущего location.pathname
  - **path** — путь в компоненте Route(должен быть абсолютным) (**v6 - принимает массив**)
  - **isExact** — path в Route === location.pathname
  - **params** — объект содержит значения из path которые возвращает модуль path-to-regexp.

- Можете поиграться с тестером роутов и посмотреть, как создается объект match.

- **Создание наших роутов** — 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.

- **Что делает рендер компонента 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.

- Элементу, отрендеренному Route, будет передано несколько props:

  match — объект сопоставления path с *location.pathname*,

  location объект и *history* объект (созданный самим роутом).

- **Параметры в path**. Иногда нам требуется использовать переменные для получения какой-либо информации. К примеру, роут профиля игрока, где нам требуется получить номер игрока. Мы можем сделать это, добавив параметр в prop path. path='/roster/:number' - :number часть строки в /roster/:number означает, что часть path после /roster/ будет получена в виде переменной и сохранится в match.params.number.

CyberBionic systematics

CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

К примеру путь /roster/6 сгенерирует следующий объект с параметрами: ({ number: '6' // Любое переданное значение интерпретируется как строка }).

- **Ссылки** — наш сайт нуждается в навигации между страницами. Если мы создадим обычные ссылки, то страница будет перезагружаться. React Router решает эту проблему компонентом <Link/> который предотвращает перезагрузку. Когда мы кликаем на <Link/> он обновляет URL и React Router рендерит нужный компонент без обновления страницы.
- **<Link/>** - использует prop *to* для описания URL куда следует перейти. Prop to может быть строкой или location объектом (который состоит из pathname, search, hash, state свойств). Если это строка, то она конвертируется в location объект. Пути в компонентах <Link/> должны быть абсолютными.
- **Location object -** Объект location описывает разные части URL'a. { pathname: '/', search: '', hash: '', key: 'abc123' state: {} }.
- **React Router Config** - библиотека, помощник настройки статического маршрута для React Router. Данная библиотека позволяет настраивать роутинг в виде объекта. (не является обязательным к использованию вместе с библиотекой «react-router»).

Закрепление материала

- Какие props принимает компонент <Route>?
- Какие props принимает компонент <Link>?
- Какая разница между props render и children в компоненте Route?

Дополнительное задание

Задание
Посмотрите основные API и примеры использования react-router.

Самостоятельная деятельность учащегося

Задание 1
Выучите основные понятия, рассмотренные на уроке.

Задание 2
Создайте маршрутизацию в проекте с помощью react-router-config.

Задание 3
Перепишите маршрутизацию, созданную с помощью react-router-config с использованием компонентов <Route> и <Switch>, с возможностью выбирать разные layout для страницы.

Рекомендуемые ресурсы

react-router-github
reacttraining.com
туториал React Router v4
React Router v5
туториал React Router v6
path-to-regexp

CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com