

# Написание полноценного SPA

**№ урока:** 6 **Курс:** Написание полноценного SPA

**Средства обучения:** Текстовый редактор или IDE, браузер

## Обзор, цель и назначение урока

В этом уроке мы рассмотрим концепцию “умных” и “глупых” компонентов, отображение ошибок в интерфейсе, отображение индикаторов загрузки, оптимистичные действия, конфигурацию и создадим финальный build нашего приложения для работы с задачами.

## Изучив материал данного занятия, учащийся сможет:

- Делить компоненты приложения на “умные” и “глупые”
- Настраивать конфигурацию
- Настраивать отображение индикаторов загрузки
- Отображать ошибки в приложении

## Содержание урока

1. Концепция “умных” и “глупых” компонентов
2. Пример. Деление компонентов на “умные” и “глупые”
3. Пример. Отображение индикатора загрузки для списка задач
4. Что такое оптимистичные действия и когда стоит их применять?
5. Пример. Оптимистичное выделение задачи
6. Пример. Оптимистичное редактирование задачи
7. Пример. Отображение ошибки при неправильном id списка задач
8. Конфигурация, разные подходы
9. Пример. Создаем конфиг в нашем приложении
10. Пример. Работа с .gitignore
11. Пример. Финальная сборка

## Закрепление материала

1. Как делятся компоненты на “умные” и “глупые”?
2. Чем они принципиально отличаются?
3. Какие преимущества такого подхода?
4. Какие подходы к конфигурации можно использовать?
5. Какая команда отвечает за сборку приложения?
6. Как работать с .gitignore?

## Самостоятельная деятельность учащегося

### Задание 1: Оптимистичное редактирование списка задач

*Уровень сложности: низкий*

При изменении названия списка задержки тоже быть не должно.

### Задание 2: Отображение ошибки при создании задачи

*Уровень сложности: низкий*

При неверном введении данных для нового списка задач (так же и при редактировании) должна появляться ошибка.

### Задание 3: Централизованное отображение ошибок

*Уровень сложности: высокий*

Сделайте централизованное отображение ошибок.

## Задание 4: Обработка неудачи при оптимистичных действиях

Уровень сложности: высокий

При неудачном выполнении действия, изменения, внесенные пользователем, должны отменяться.

### Рекомендуемые ресурсы

Все примеры из урока и пояснения к домашним заданиям можно найти тут:

<https://github.com/krambertech/react-essential-course>

#### Flux

- Официальный сайт - (<https://facebook.github.io/flux/>)

#### Google API

- <https://console.developers.google.com/iam-admin/projects>
- <https://developers.google.com/google-apps/tasks/v1/reference>

#### Дополнительно

- Библиотека material-ui (<http://material-ui.com>)

#### Что почитать:

Официальная документация React - <http://facebook.github.io/react/docs>

Советую ознакомиться с JavaScript Style Guide - <https://github.com/airbnb/javascript>