



Разработка ПРИЛОЖЕНИЙ

Занятие #6. API

Прогресс

- ★ create-react-app, ОСНОВЫ, npm
- ★ КОМПОНЕНТЫ, jsx
- ★ git
- ★ Состояния, переменные, props
- ★ useState()
- ★ Стилизация
- ★ useEffect(), useRef()
- ★ Отладка
- ★ Маршруты, React Router
- ★ Redux, useContext()

☐ Firebase и API

☐ Авторизация

☐ База данных

☐ Хостинг

☐ Деплой

☐ БЭМ, цикл проекта, figma

☐ PWA

☐ Улучшения, библиотеки

☐ React + TS

☐ Next.js

ПЛАН

01

REST API

02

Запросы в React

03

Firebase



01 Rest API

HTTP API

- Работает на HTTP
- Возвращает JSON/XML (SOAP)
- Endpoint = URL

Примеры

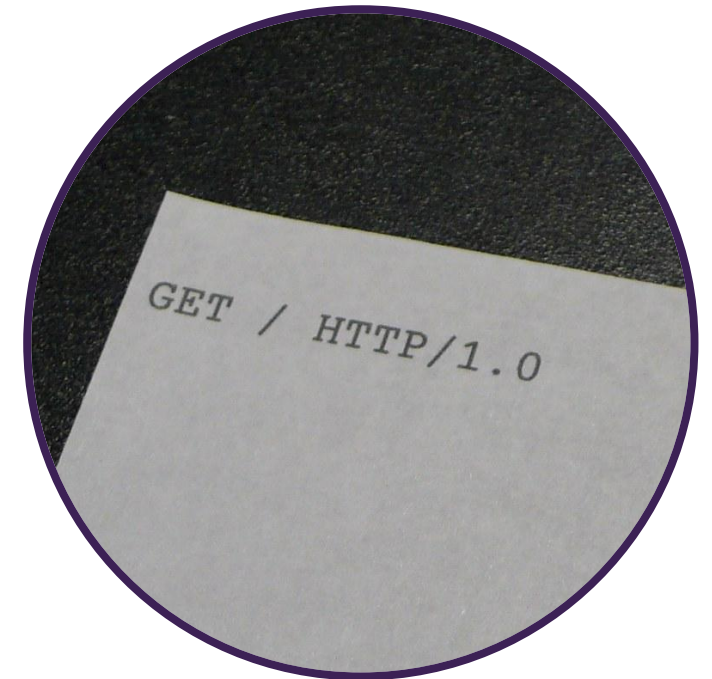
<https://api.github.com/users>

<http://api.bitcoincharts.com/v1/markets.json>

[Подборка публичных API](#) и [ещё одна](#)

[Статья на хабре](#)

ПО для тестирования API: [https://www.postman.com](https://www.postman.com/postman.com/downloads)
[postman.com>downloads](https://www.postman.com/postman.com/downloads)



REST (Representational State Transfer)

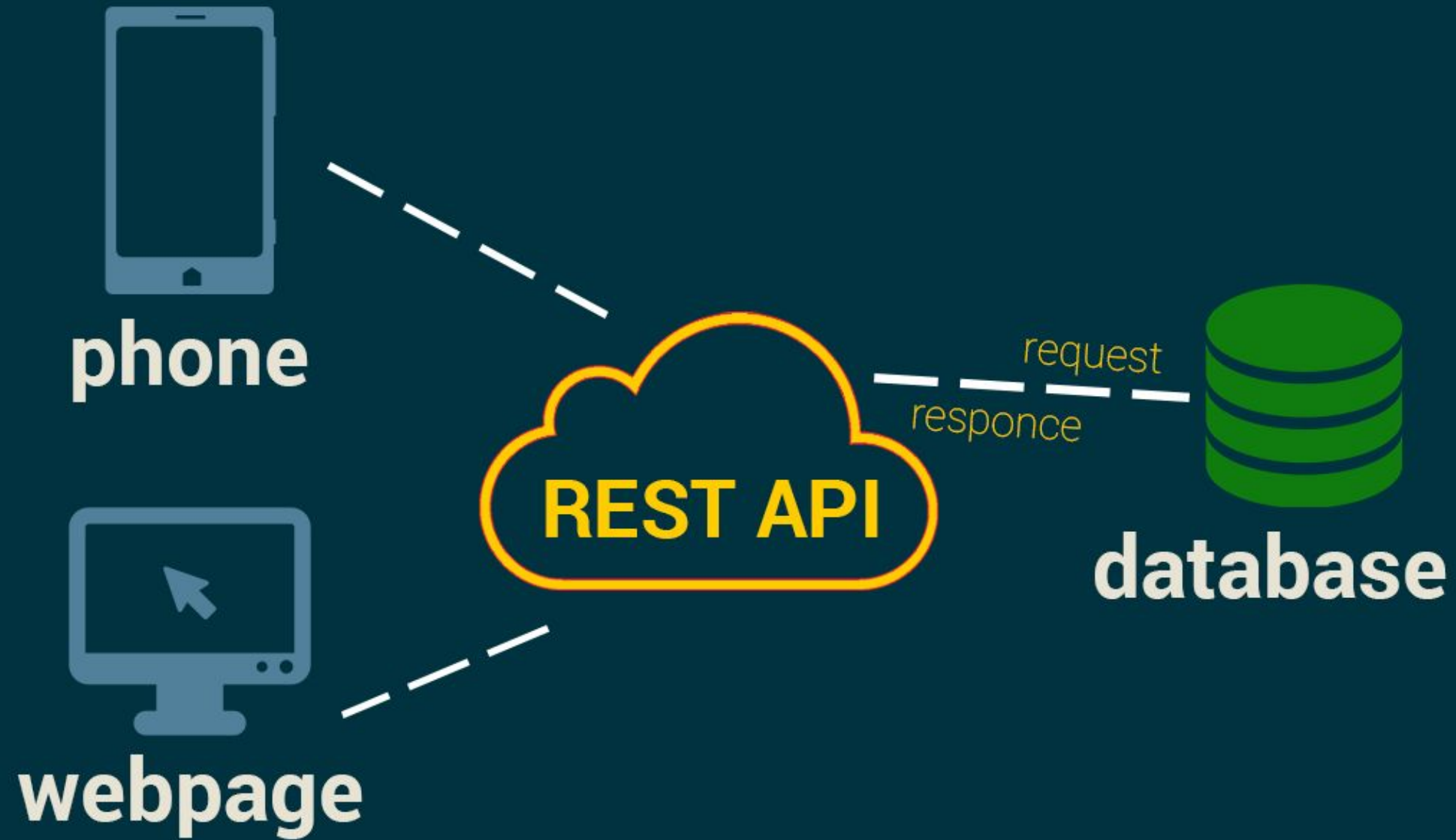
- **Архитектурный стиль** взаимодействия компонентов распределённого приложения

Требования

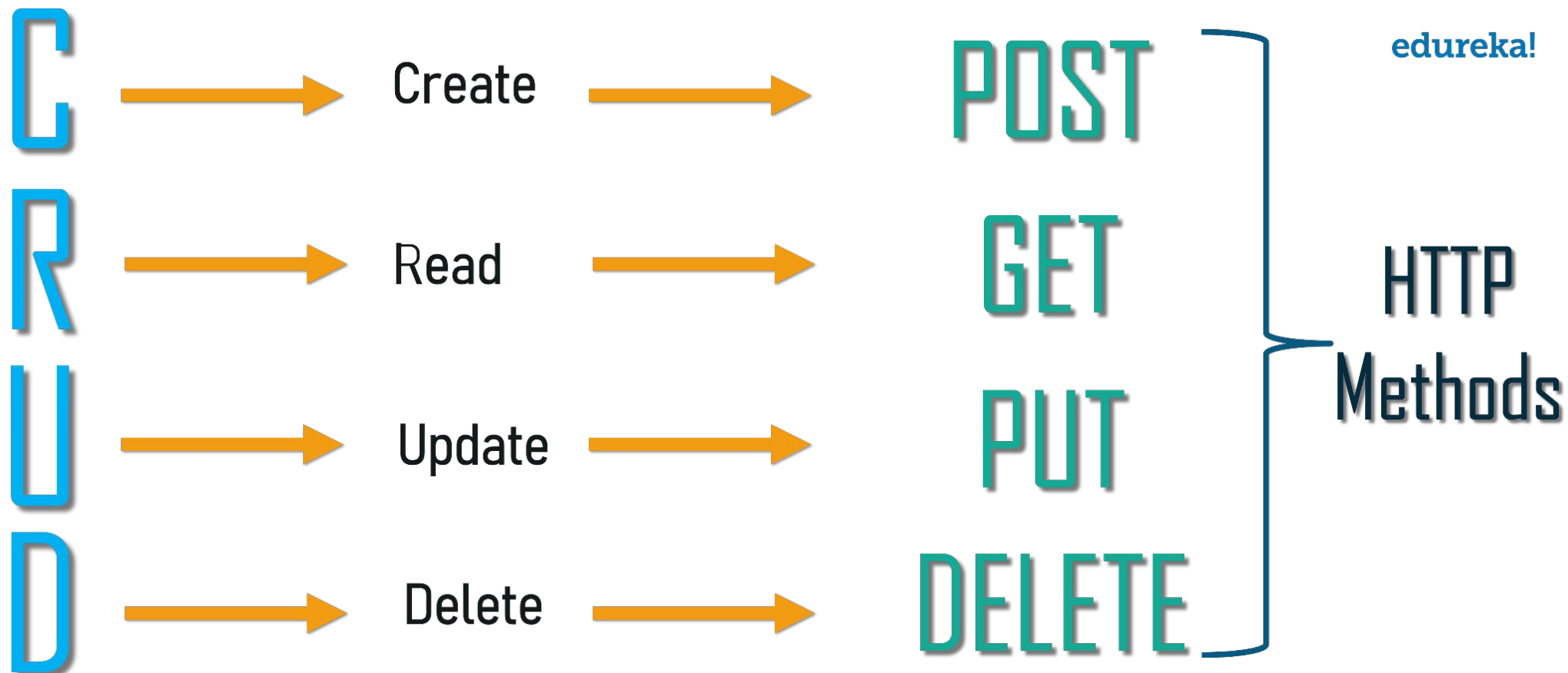
- Клиент-серверная модель (client-server model).
- Отсутствие состояния (statelessness).
- Кэширование (cacheability).
- Единообразие интерфейса (uniform interface).
- Многоуровневая система (layered system).
- Код по требованию (code on demand) — необязательно.



Рой Филдинг, автор REST.



Методы в HTTP API

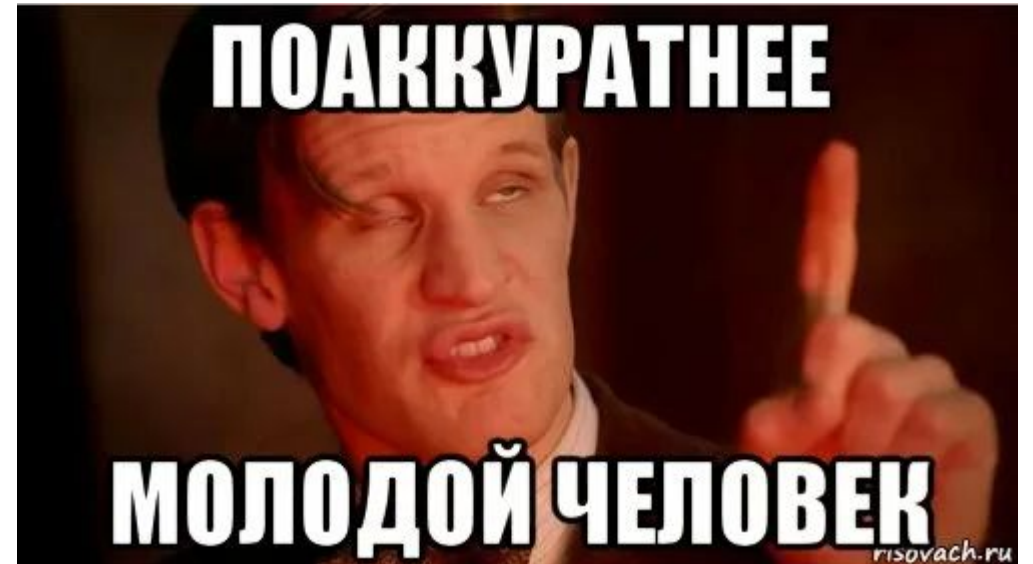




02 Запросы в React

Как не надо делать

- запросы к базе напрямую в коде React
- хранить пароли в репозитории
(а как? - напр. [ansible](#), [GitHub secrets](#))
- API без авторизации



Работа с внешним сервером в React

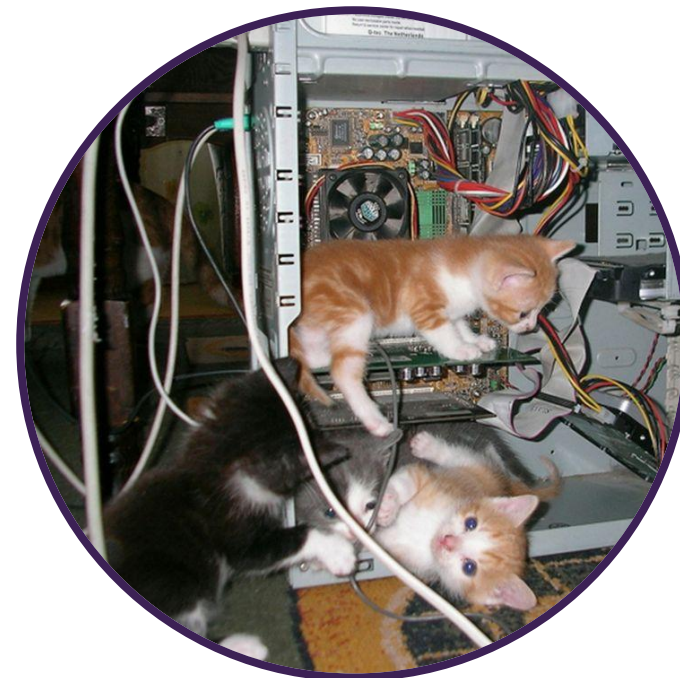
- Собственный сервер (node.js, python, PHP и т. д.)
- Публичные API-сервисы
- Готовые SaaS ([firebase](#), [parse](#), [back4app](#))
- статические файлы .json на хостинге

Работа с внешним сервером в React

Пример проекта

Почитать про fetch()

Почитать про async-await

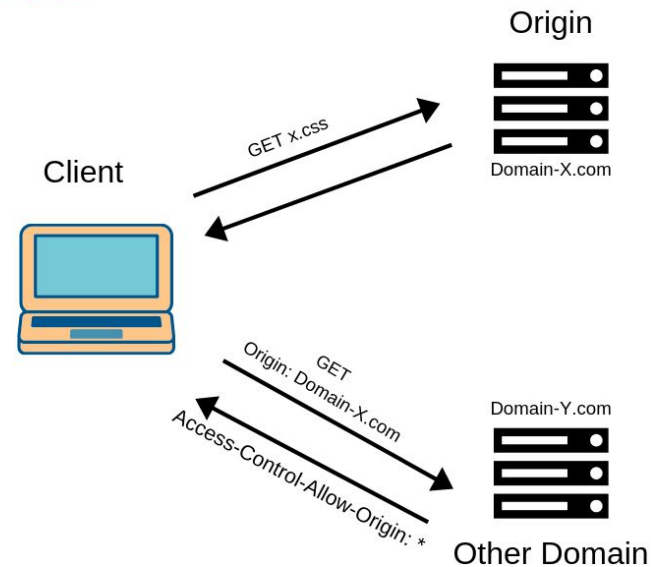


CORS

✗ Access to fetch at '<https://www.youtube.com/watch?v=9QneqUhCVtU>' from origin '<http://localhost:3000>' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource. If an opaque response serves your needs, set the request's mode to 'no-cors' to fetch the resource with CORS disabled.

[статья](#) и [Статья](#)

CORS





03 Firebase

Firebase

firebase.google.com

- Авторизация
- База данных
- Хранение файлов
- Хостинг
- Аналитика



Начало работы

firebase.google.com

- <https://firebase.google.com/docs>
- Создать новый проект
- Project Overview - Register Web App
- `npm install firebase`
- создать файл конфигурации
(добавить в gitignore)



Работа с авторизацией в firebase

firebase.google.com

Урок на YouTube

<https://www.youtube.com/watch?v=PKwu15ldZ7k>

