МШП 20.35 УНИВЕРСИТЕТ КОД БУДУЩЕГО Минцифры России НАЦИОНАЛЬНЫЕ ПРОЕКТЫ РОССИИ ЦИФРОВАЯ ЭКОНОМИКА

Fullstack-разработка

NodeJS
GET-запросы

МШП

Цели урока:

1. Будем отправлять на сервер полезные данные
2. Научим сервер возвращать HTML + CSS + JS

МШП

Разбор ДЗ

Остались ли вопросы?

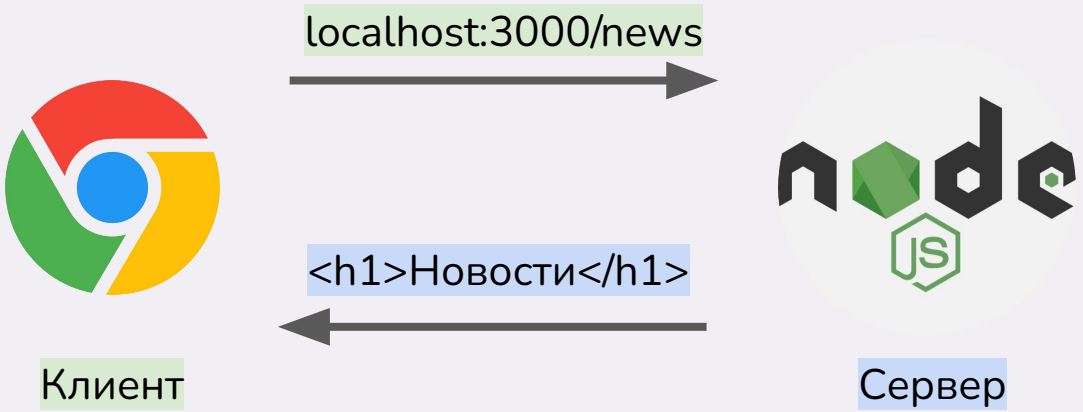
МШП

Теория

Запрос с параметром

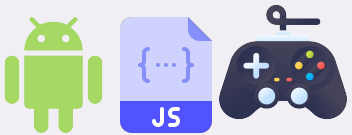
МШП

Давайте представим, что наш сервер публикует главные новости в сфере IT.



МШП

Наш сайт посещают самые разные категории пользователей:



localhost:3000/news

<h1>Новости</h1>

Сервер

МШП

Если в один момент времени всем показывать одинаковые новости, то кому-то будет не интересно

МШП

Поэтому клиент может вместе
с запросом передать дополнительную
информацию о себе

МШП

URL-адрес

```
http://localhost:3000/news?topic=js
```

Через **URL-адрес** клиент составляет запрос на сервер.

Протокол, доменное имя и порт олицетворяют собой физический адрес **сервера в сети.**

МШП

URL-адрес

```
http://localhost:3000/news?topic=js
```

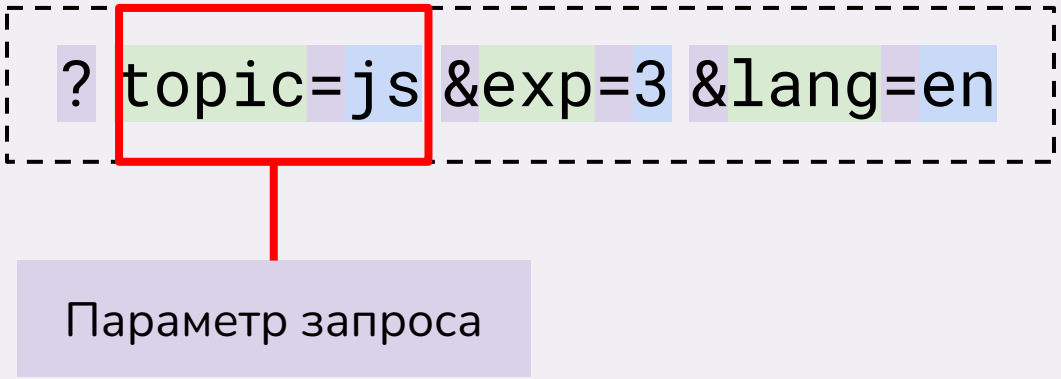
Оставшуюся часть URL уже обрабатывает сервер:

Путь – это **роут** на сервере, который будет обрабатывать запрос пользователя.

Параметр – это информация, которая будет использована для составления ответа.

МШП

Параметры запроса



Запрос очень похож на объект в JS.

Он состоит из **ключей** и **значений**.

Пара из ключа и значения называется **параметр запроса или query.**

МШП

req.query

```
app.get(`/news`, function (req, res) {
    let topic = req.query.topic;

    res.send(`
        <h1>Новости по теме ${topic}</h1>
    `);
});
```

/news?topic=js

Внутри роута можно прочитать значение параметра **query** из **запроса** и использовать его для создания **ответа сервера**.

МШП

req.query

```
app.get(`/news`, function (req, res) {
    let exp = Number( req.query.exp );

    res.send(`
        <h1>Ваш опыт работы ${exp} лет</h1>
    `);
});
```

/news?exp=3

Параметр – это всегда тип данных строка.

Поэтому при работе с числами обязательно используйте приведение типов через Number.

МШП

req.query

```
app.get(`/news`, function (req, res) {
    let topic = req.query.topic;

    if (!topic) {
        // параметр не передали
    } else {
        // всё ок
    }
});
```

/news

/news?topic=js

Параметр могут не передать или передать неправильно.

Поэтому **всегда добавляй проверки!**

МШП

Пример: Калькулятор

```
app.get(`/sum`, function (req, res) {
  let a = Number(req.query.a);
  let b = Number(req.query.b);

   if (!a || !b) {
        // параметры не передали
   } else {
        res.send(`Сумма: ${a + b}`);
   }
});
```

http://localhost:3000/sum?a=4&b=5

МШП

Практика

1. Запрос с параметром
2. Изображения

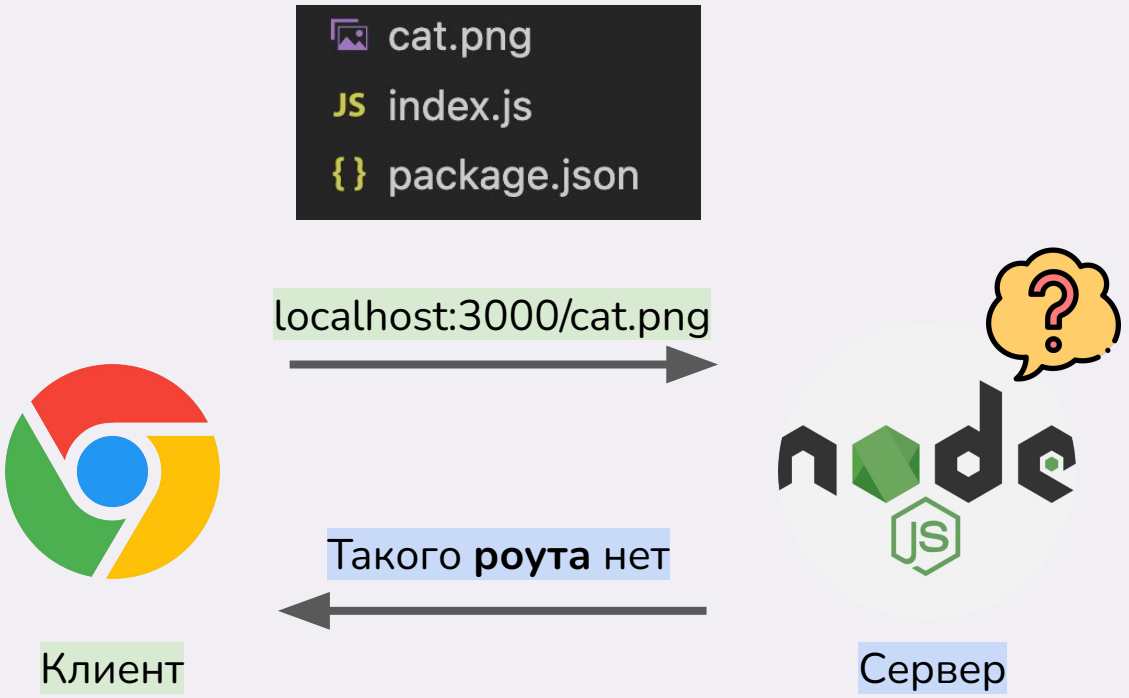
МШП

Теория

Раздача статики

МШП

Как получить с сервера изображение?



МШП

Раздача статики

```
app.get(`/cat.png`, function (req, res) {

    // Может быть, добавить роут?

});
```

Нет! Ведь если картинок много, придётся на каждую добавлять отдельный роут.

МШП

Раздача статики

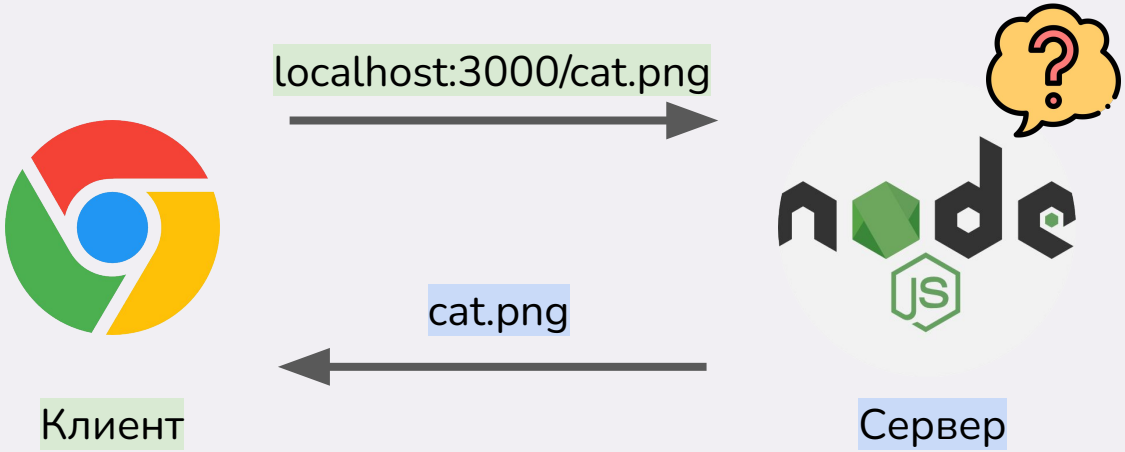
```
let express = require(`express`);
let app = express();
let port = 3000;

// Добавь эту строку в начало проекта
app.use(express.static(`public`));
```

Эта маленькая волшебная строка делает файлы из папки **public** доступными для клиента.

МШП

Теперь сервер будет искать файлы в папке public



МШП

Раздача статики

```
app.get(`/profile`, function (req, res) {
   res.send(`
     <h1>Профиль пользователя</h1>
     <img src="cat.png">
   `)
})
```

http://localhost:3000/profile

Если правильно настроить раздачу статики, то можно будет использовать изображения из папки **public** для создания HTML-страниц в роутах.

МШП

Вот такая быстрая теория!

А теперь давайте применим её к задаче про поиск изображений...

МШП

Практика

Поиск по картинкам на сервере

МШП

Сколько раз за сегодняшнее занятие вы остановили и запустили сервер?

МШП

Установка пакета

Выполни в терминале VSCode команду:

```
npm install nodemon
```

Nodemon будет перезапускать
сервер каждый раз,
когда мы сохраняем файл **index.js.**

МШП

Установка пакета

Измени package.json:

```
{
  ...
  "main": "index.js",
  "scripts": {
    "start": "nodemon index.js"
  },
  "keywords": [],
  ...
}
```

МШП

Установка пакета

Выполни в терминале VSCode команду:

```
npm start
```

```
> project1@1.0.0 start
> nodemon index.js

[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Сервер запущен: http://localhost:3000
```

Готово!

С этого момента запускай сервер с помощью этой команды.

МШП

Статические файлы

1. HTML-страницы
2. CSS-файлы
3. JS-файлы
4. Медиа-файлы
5. ... любые другие файлы ...

Теперь все эти файлы можно раздавать с помощью нашего сервера. А это значит...

CSS и Bootstrap снова с нами

МШП

Статические файлы

По какому URL можно получить файл index.html?

➔ http://localhost:3000/index.html

➔ http://localhost:3000

МШП

Статика

- В статике хранят те страницы, которые для всех пользователей выглядят одинаково

МШП

Роуты

- Используй роут, когда нужно сгенерировать HTML под запрос пользователя.
- В роутах можно использовать CSS, но это неудобно :)

МШП

Практика

Пользователи

МШП

Цели урока:

- ✓ Отправляли на сервер полезные данные через query
- ✓ Научили сервер возвращать HTML + CSS + JS

МШП

Домашнее задание