

Fullstack-разработка

NodeJS

Шаблонизация

Цели урока:

1. Сделаем универсальную страницу для нескольких роутов
2. Познакомимся с новым синтаксисом

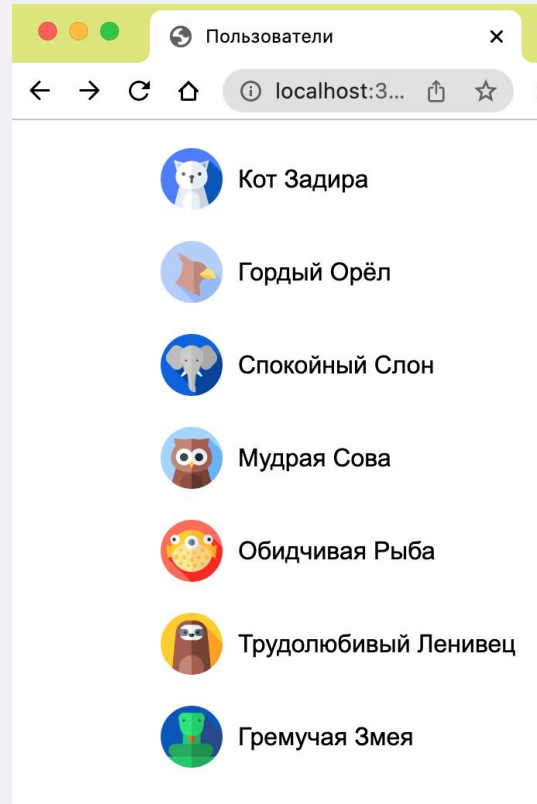
Разбор ДЗ

Как можно сделать
его красивым?

Теория

Шаблонизация

Как вывести на
страницу список
пользователей,
который хранится
на сервере?



Рендер массива

Составить вёрстку списка в JS:

```
let html = '<ul>';  
for (let i = 0; i < array.length; i++) {  
    html += '<li>${array[i]}</li>';  
}  
html += '</ul>';  
  
res.send(html);
```

Как добавить в вёрстку CSS?

Рендер массива

Вернуть клиенту страницу со стилями.

```
res.send(`
  <!DOCTYPE html>
  <html lang="ru">
  <head>
    <meta charset="UTF-8">
    <title>Пользователи</title>
    <link href="css/users.css"
      rel="stylesheet">
  </head>
  <body>${html}</body>
</html>
`);
```

Формировать HTML через JS можно,
но оно того не стоит.

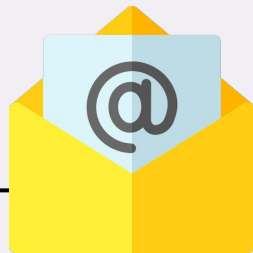
Почему?

- Нет подсветки синтаксиса
- Слишком много кода
- Тяжело вносить изменения



Решение есть!

Шаблон – HTML-код, в котором заранее отведено место для данных.



```
<h1>Уважаемый {{ username }}!</h1>  
<p>Заказ №{{ order }} уже ждёт вас в  
пункте выдачи {{ address }} </p>
```

Решение есть!

Шаблонизатор – программа, которая подставляет в шаблон реальные данные.



```
<h1>Уважаемый Михаил!</h1>  
<p>Заказ №1738920 уже ждёт вас в  
пункте выдачи г Москва, ул  
Доставочная, д 1 </p>
```

Установка пакета

Handlebars это популярный [шаблонизатор для Express](#) проектов. Он сильно проще, чем его аналоги.

Выполни в терминале VSCode команду:

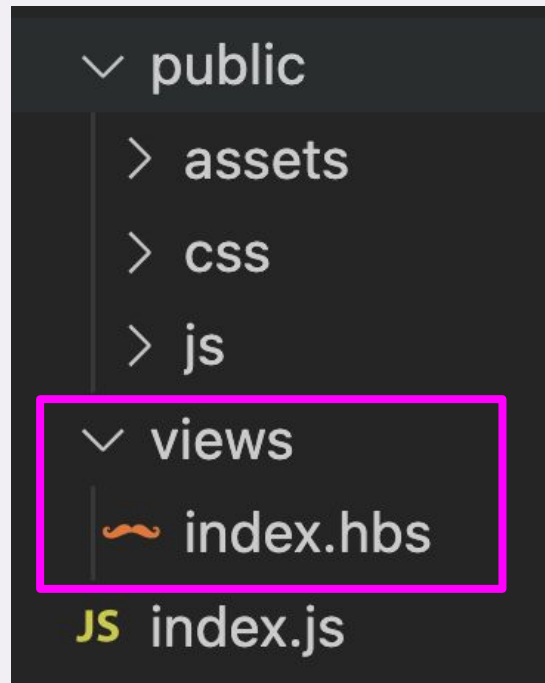
```
npm install hbs
```



Структура проекта

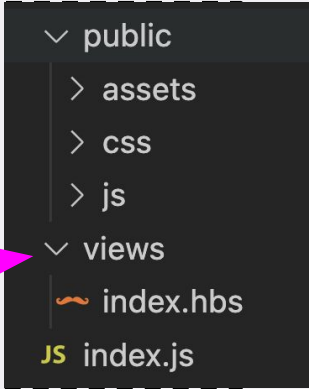
Теперь в файлах
статике останутся
только стили,
изображения и js.

А весь HTML-код
переедет в папку **views**
с новым расширением
hbs



Настройка Express

```
let express = require(`express`);  
let app = express();  
  
const hbs = require('hbs');  
app.set('views', 'views');  
app.set('view engine', 'hbs');
```



File explorer showing project structure:

- public
 - assets
 - css
 - js
- views
 - index.hbs
 - index.js

A pink arrow points from the `app.set('views', 'views');` line in the code to the `views` directory in the file explorer.

Теперь всё готово!

Давайте посмотрим, как это всё будет работать.



Рендер шаблона

Помните, раньше мы писали с вами функции `renderItem`, `renderAll`?

На сервере **render** – это процесс создания HTML файла для клиента

Рендер шаблона

Для **отрисовки шаблона**
просто укажите **имя файла**.

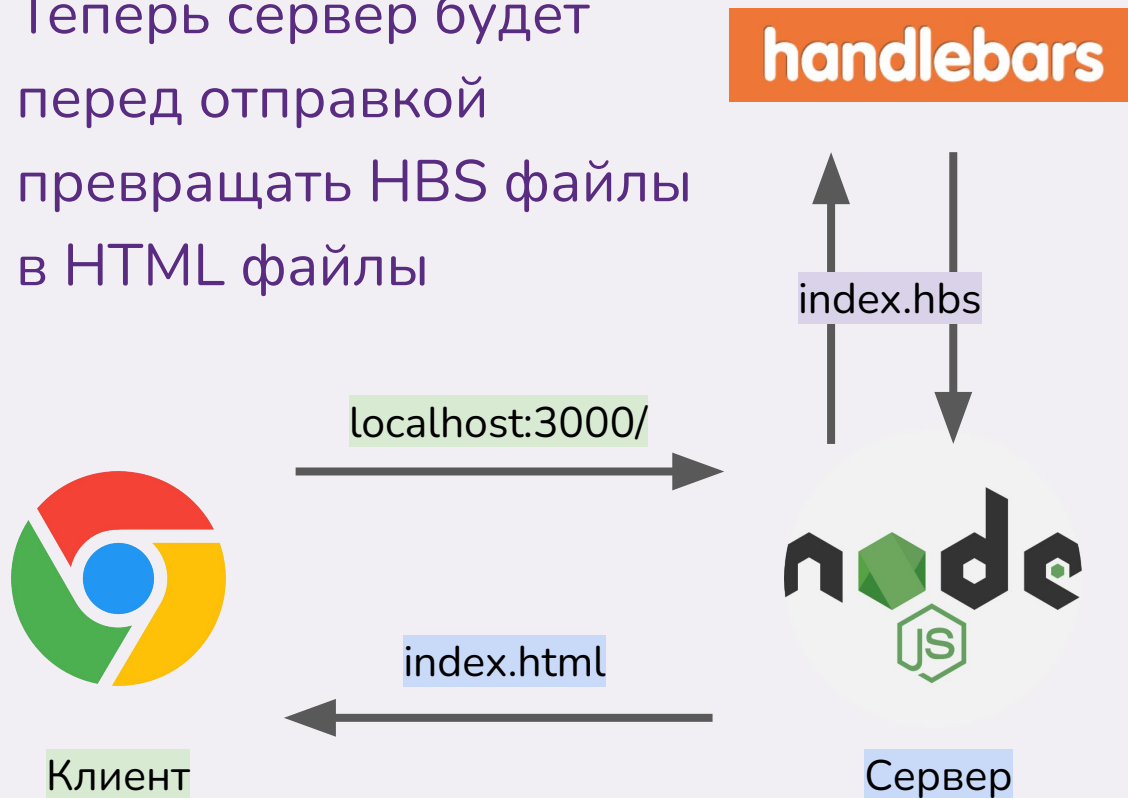
```
<h1>Привет, мир!</h1>
```

index.hbs

```
app.get('/', function (req, res) {  
  res.render('index');  
})
```

index.js

Теперь сервер будет
перед отправкой
превращать HBS файлы
в HTML файлы



Рендер данных

Функция `render` принимает объект с данными

```
app.get(`/`, function (req, res) {  
  res.render(`index`, {  
    username: 'Королева Англии',  
    age: 95,  
  });  
})
```

index.js

```
<p>Имя: {{username}}</p>  
<p>Возраст: {{age}}</p>
```

index.hbs

Итоги

- Шаблоны и JS файлы будут существовать отдельно
- В шаблоне используется обычный HTML-код и `{{ }}` для подстановки данных
- **Один шаблон можно переиспользовать для разных данных**

Практика

АрхиРыба

Теория

Хелперы

Язык шаблона отличается от
обычного HTML специальными
возможностями:

1. **Данные:** `{{ username }}`
2. **Условия:** `<#if> </if>`
3. **Циклы:** `<#each array> </each>`

Условие if

IF убирает лишние элементы из шаблона.

```
app.get(`/`, function (req, res) {  
  res.render(`index`, {  
    username: 'Королева Англии',  
    image: 'queen.jpg',  
  });  
})
```

index.js

```
{{#if image}}  
    
{{/if}}
```

index.hbs

Условие if

IF убирает лишние элементы из шаблона.

```
app.get(`/`, function (req, res) {  
  res.render(`index`, {  
    showComments: false,  
  });  
})
```

index.js

```
{{#if showComments}}  
  ...вёрстка комментариев...  
{{/if}}
```

index.hbs

Цикл each

Как нарисовать повторяющиеся элементы?

```
res.render(`index`, {  
  array: ['cat', 'net', 'sad']  
});
```

index.js

```
<ul>  
  {{#each array}}  
    <li>{{@index}}. {{this}}</li>  
  {{/each}}  
</ul>
```

index.hbs

Цикл each

Шаблон:

```
<li>{{@index}}. {{this}}</li>
```

Результат:

```
<ul>  
  <li>0. cat</li>  
  <li>1. net</li>  
  <li>2. sad</li>  
</ul>
```

Цикл each

Как нарисовать массив объектов?

```
res.render(`index`, {  
  array: [  
    {username: 'tester'},  
    {username: 'admin'}  
  ]  
});
```

index.js

```
{{#each array}}  
  <div>{{this.username}}</div>  
{{/each}}
```

index.hbs

Практика

Новости Разработки

Алгоритм шаблонизации

- Подготовить вёрстку на HTML
- Превратить вёрстку в шаблон через `{{ }}`, `if`, `each`
- Передать данные в шаблон через метод `render`

Цели урока:

- ✓ Сделали универсальную страницу для нескольких роутов
- ✓ Познакомились с новым синтаксисом (и технологией!)

Домашнее задание