

Fullstack-разработка

Vue

Роутинг

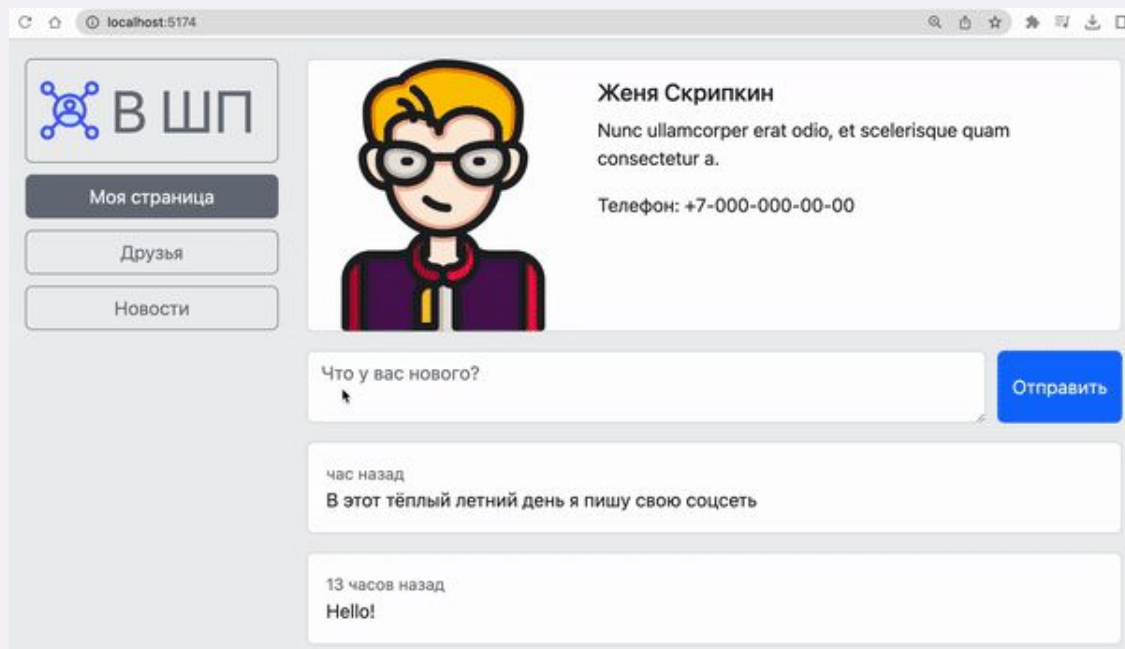
Цель урока:

Создать многостраничное
приложение на Vue

Разбор ДЗ

Что получилось?

Социальная сеть



Практика

Профиль пользователя

Теория

Клиентский роутинг

Роутинг – это организация перехода по ссылкам между страницами сайта

```
<a href="/login">  
    Войти  
</a>
```

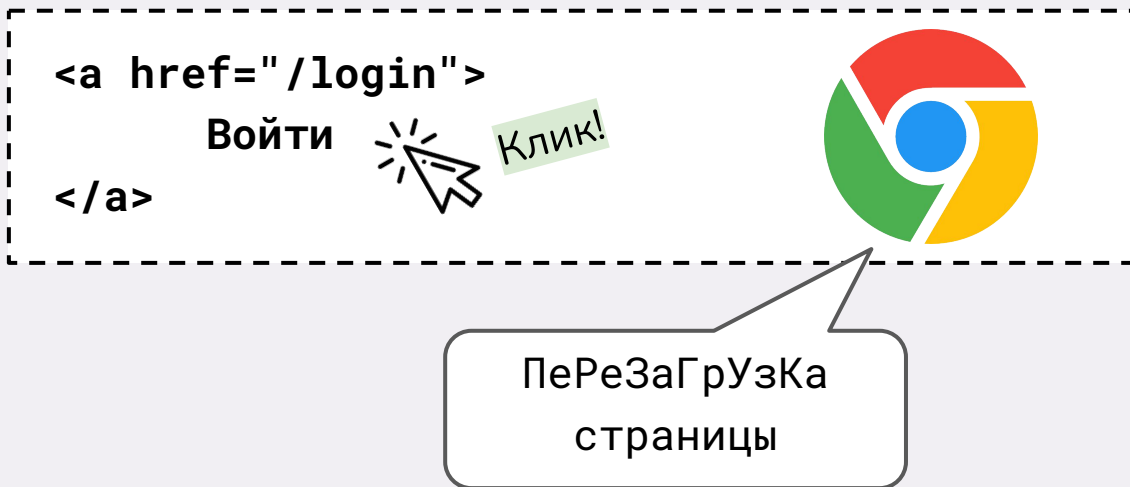


Ранее мы реализовывали
серверный роутинг силами
Express и Handlebars

```
/          -> index.hbs  
/login     -> login.hbs  
/profile   -> profile.hbs
```



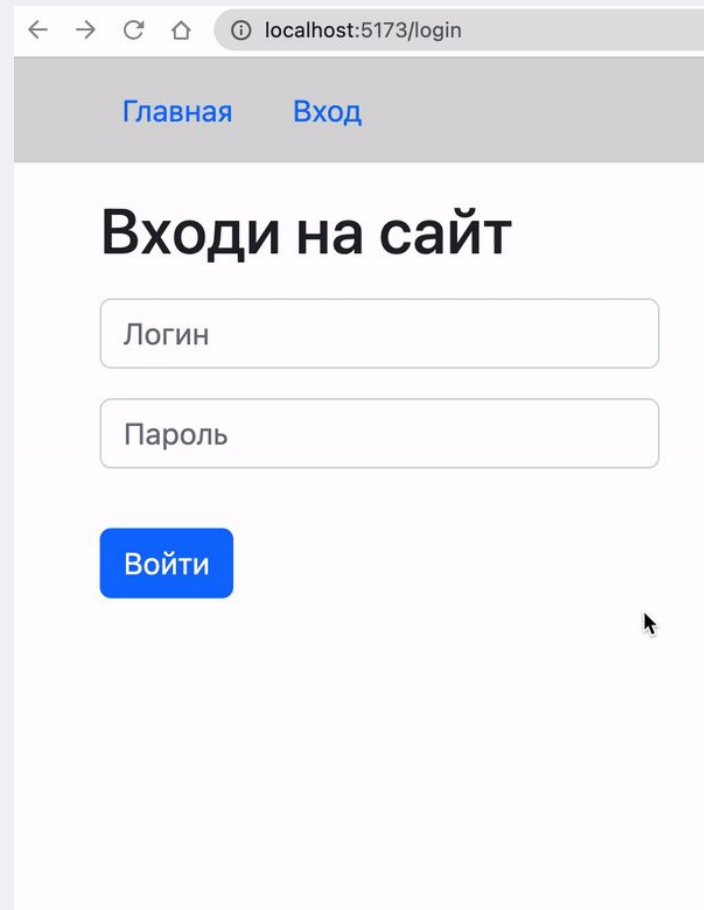
У серверного роутинга есть одна большая проблема...



Клиентский роутинг – это навигация
внутри браузера без перезагрузки
страницы с помощью **JS**



Эту задачу отлично
решает библиотека
vue-router



← → ↻ 🏠 ⓘ localhost:5173/login

[Главная](#) [Вход](#)

Входи на сайт

Логин

Пароль

Войти

Обычный компонент, не меняется

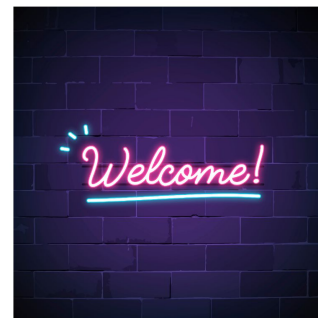
[Главная](#)[Вход](#)

Компоненты-страницы, меняются

Входи на сайт

LoginPage

Добро пожаловать!

*WelcomePage*

router.js

В файле **router.js** хранится связь между **путём** и его **компонентом-страницей**

```
routes: [  
  {  
    name: 'login',  
    path: '/login',  
    component: LoginPage  
  },  
  {  
    name: 'home',  
    path: '/home',  
    component: HomePage  
  }  
]
```

localhost:5173/login

- название
- путь
- компонент

router-view

Роутер встраивается в главный компонент приложения **App.vue**

```
import { RouterView } from 'vue-router';  
  
export default {  
  components: { RouterView }  
}
```

```
<header> Шапка сайта </header>  
<router-view></router-view>  
<footer> Подвал </footer>
```

И последнее...

Как же теперь будут работать ссылки?

```
<a href="/login">
```

Войти

```
</a>
```



Клик!



Навигация

Внутри любых компонентов теперь доступен специальный объект **\$router**, который по названию **роута** покажет нужный компонент

```
methods: {  
  goLogin() {  
    this.$router.push({  
      name: 'login'  
    })  
  }  
}
```

```
<div @click="goLogin">Вход</div>
```


Навигация

Также доступен объект `$route`,
содержащий настройки открытого пути.

Его можно использовать внутри шаблона
или методов как обычное **data-свойство**.

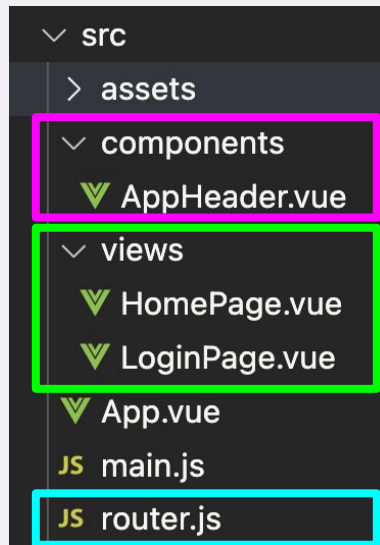
```
mounted() {  
  console.log(this.$route);  
}
```

```
<div> {{ $route.path }} </div>
```

```
fullPath: "/login"  
hash: ""  
href: "/login"  
▶ matched: [{...}]  
▶ meta: {}  
name: "login"  
▶ params: {}  
path: "/login"  
▶ query: {}
```

Структура проекта

Сейчас нам предстоит внести много изменений в проект, не будем терять времени!



Обычные компоненты

Компоненты-страницы

Роутер

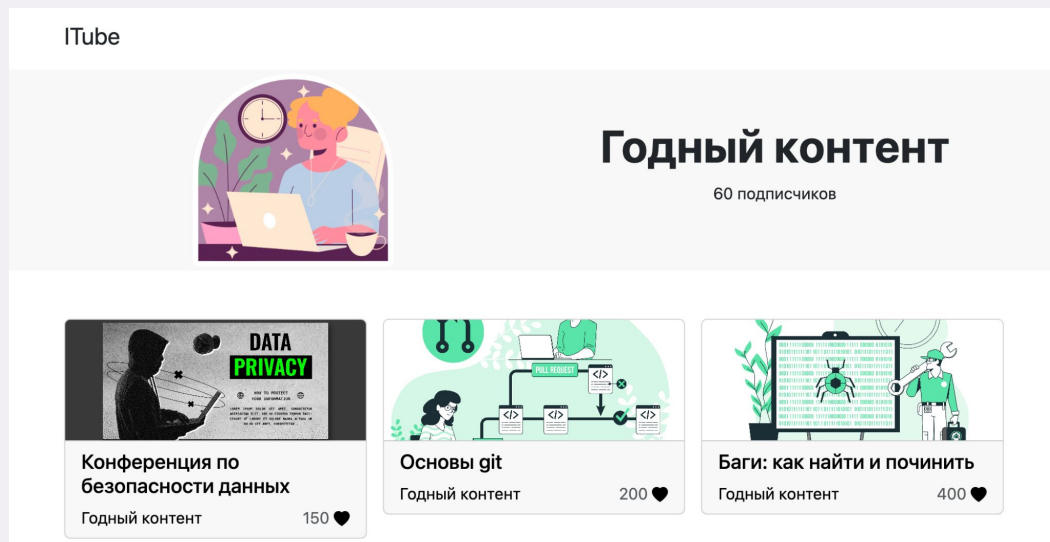
Практика

Две страницы

Теория

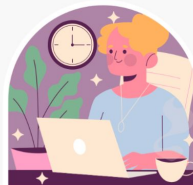
Динамические пути

Знакомый проект?



Какая технология позволила нам показывать разные данные в одном и том же шаблоне?

ITube



Годный контент

60 подписчиков



Конференция по безопасности данных

Годный контент

150 ❤️



Основы git

Годный контент

200 ❤️



Баги: как найти и починить

Годный контент

400 ❤️

Параметры запроса

`http://localhost:3000/blogger?id=1234`

```
app.get(`/blogger`, function (req, res) {  
  // Получение параметра из роута  
  let id = req.query.id;  
  
  // Загрузка данных  
  let item = await Blogger.findOne({_id: id})  
  
  // Передача данных в шаблон  
  res.render('page', { blogger: item })  
});
```

Динамический путь

В vue-router можно создать **динамический путь**, который будет передавать **параметр** в компонент:

`http://localhost:5173/blogger/1234`

```
routes: [  
  {  
    name: 'blogger',  
    path: '/blogger/:id',  
    component: BloggerPage  
  }  
]
```


Динамический путь

Помните объект `$route`?

Из него мы сможем легко достать переданный **параметр** внутри компонента.

`http://localhost:5173/blogger/1234`

```
methods: {  
  async loadData() {  
    console.log(this.$route.params.id);  
    ...  
  }  
}
```

Навигация

Через объект `$router` можно перейти на роут по его **названию** и передать в него **параметр**

```
methods: {  
  goBlogger() {  
    this.$router.push({  
      name: 'blogger',  
      params: {  
        id: '1234'  
      }  
    })  
  }  
}
```

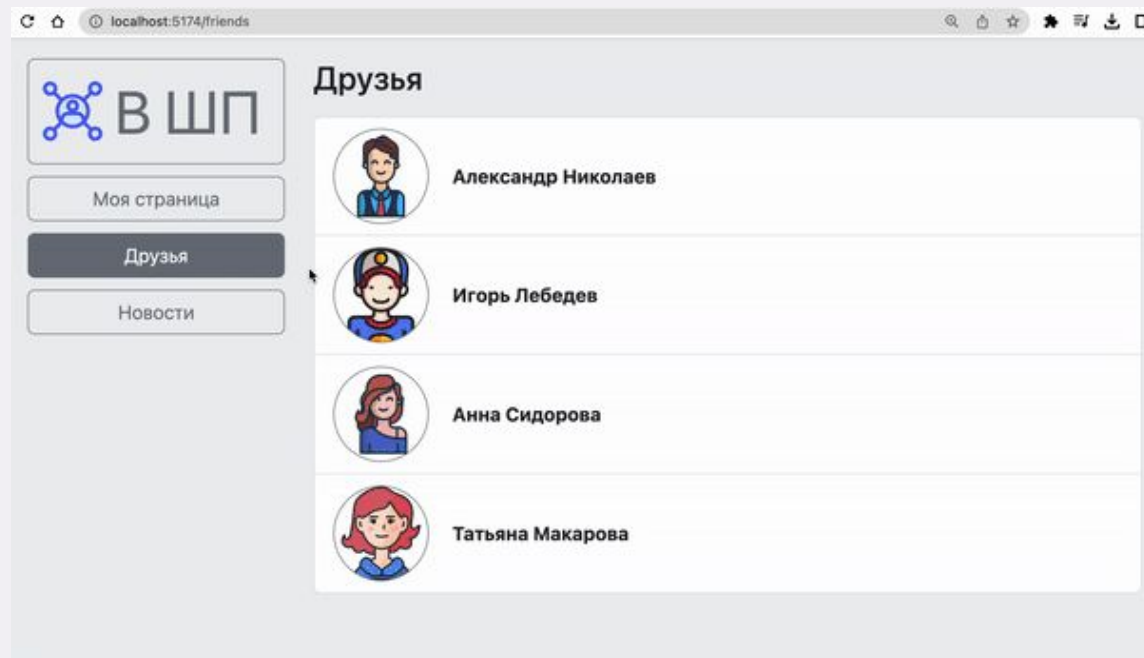
`http://localhost:5173/blogger/1234`

Динамический путь

А чтобы перезагружать данные,
когда меняется роут,
есть специальный метод **watch**:

```
data() {...},  
mounted() {...},  
methods: {...},  
watch: {  
  $route() {  
    console.log(`  
      Вызываюсь при изменении пути!  
    `);  
    this.loadData();  
  }  
}
```

Вперёд, к практике!



Практика

Такие разные пользователи

Цели урока:

- ✓ Создали многостраничное приложение на Vue

Домашнее задание