

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»  
(БГТУ им В. Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных  
систем

Лабораторная работа №6  
Дисциплина: Технологии WEB-программирования  
**«Работа с HTTP запросами»**

Выполнил: ст. группы ПВ-41

Чубов Г.С.

Проверил: Картамышев С. В.

Белгород, 2021

**Цель:** изучить принципы работы отправки Ajax запросов. Изучить работу отправки синхронных и асинхронных запросов в Vue JS. Получить навыки работы с промисами.

**Задание к лабораторной работе:**

1. Изучить возможности Vue js для отправки http запросов.
2. Выбрать подходящую библиотеку для работы с запросами.
3. Реализовать взаимодействие фронтенда с REST API, спроектированном в прошлой лабораторной работе.
4. Продемонстрировать работу взаимодействия фронтенд приложения с REST API.

## Код взаимодействия фронтенд приложения с REST API:

### Api.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\ProductController;
use App\Http\Controllers/LibraryController;
use App\Http\Controllers\UserController;

/*
|-----
| API Routes
|-----
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| is assigned the "api" middleware group. Enjoy building your API!
|
*/

Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
    return $request->user();
});

Route::prefix('products')->group(function() {
    Route::get('/', [\App\Http\Controllers\ProductController::class, 'list']);
    Route::get('/{id}', [\App\Http\Controllers\ProductController::class, 'info']);
});

Route::prefix('library')->group(function() {
    Route::post('/', [\App\Http\Controllers/LibraryController::class, 'addToLibrary']);
});

Route::prefix('user')->group(function() {
    Route::post('/', [\App\Http\Controllers\UserController::class, 'authorization']);
});
```

### Authorization

```
<script>
export default {
  name: "login",
  data() {
    return {
      userId: null,
      username: null,
      password: null,
      message: '',
      correct: false
    }
  },
  methods: {
    authorization() {
      const params = {
        name: this.username, password: this.password
      }
      this.$http.post('/user', params)
        .then(response => this.checkCorrect(response.data.userId))
    },
    checkCorrect(data) {
      if(data !== -1)
      {
        this.userId = data
        this.correct = true
        this.$router.push('/')
        this.message = ""
        localStorage.userId = this.userId
        localStorage.username = this.username
      }
    }
  }
}
```

```

        }
        else
        {
            this.message = "Неверный логин или пароль."
        }
        console.log(this.correct)
    }
},
mounted() {
    if (localStorage.userId) {
        this.userId = localStorage.userId;
    }
}
}
</script>

```

## Header

```

<script>
export default {
    name: "header",
    data() {
        return {
            user visible: false,
            login visible: true,
            username: null
        }
    },
    mounted() {
        if (localStorage.username !== null) {
            this.username = localStorage.username
            this.user visible = !this.user visible
            this.login_visible = !this.login_visible
        }
    }
}
</script>

```

## ProductPage

```

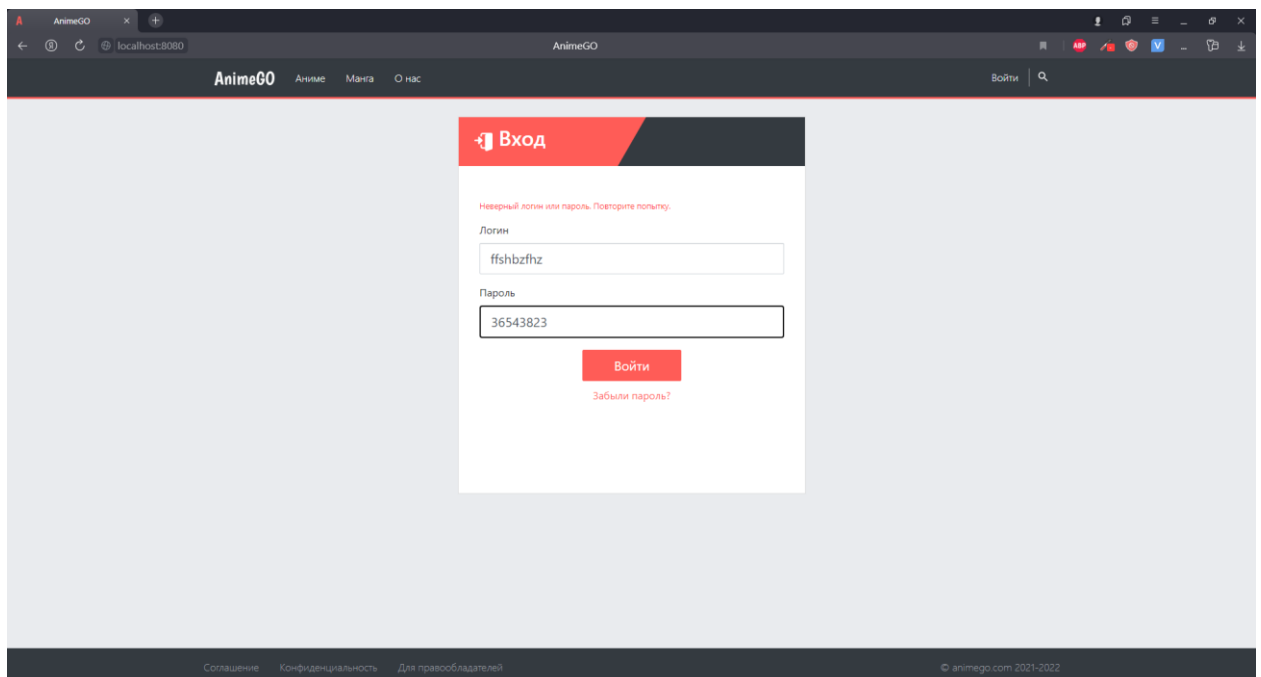
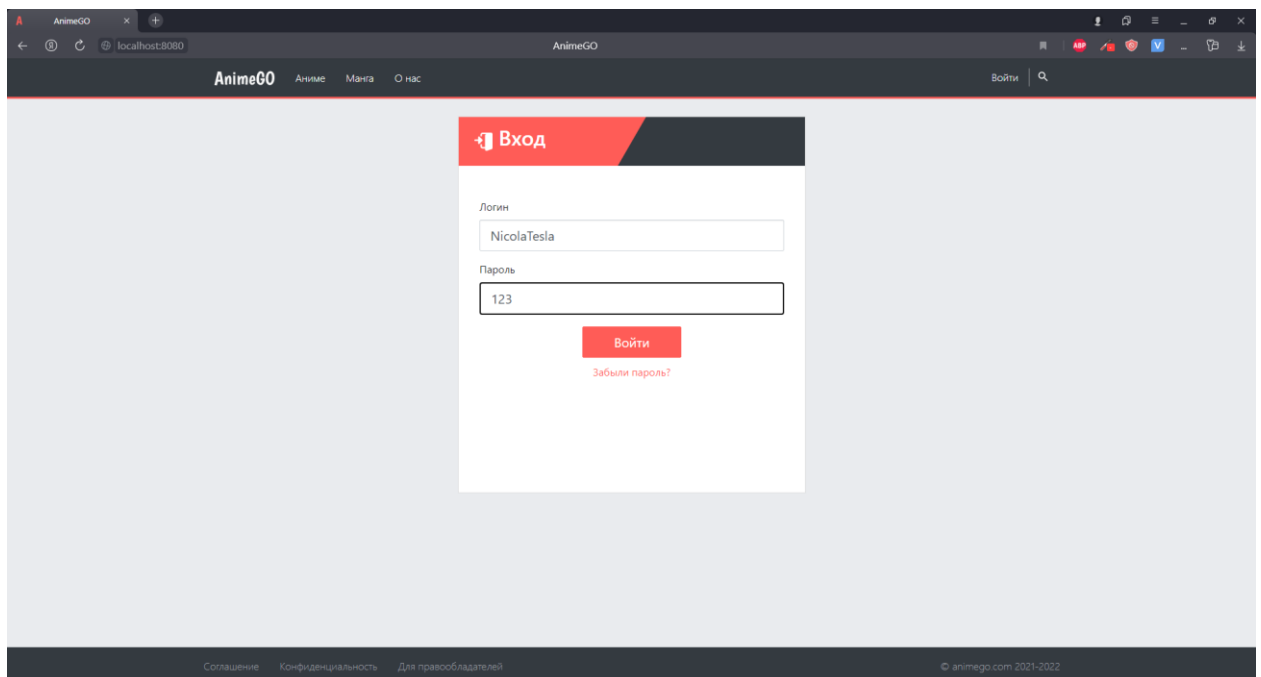
<script>
import product from "@/components/product-item"
export default {
    name: "product",
    components: {
        // eslint-disable-next-line vue/no-unused-components
        product,
    },
    methods: {
        addToLibrary(product) {
            const params = {
                id: product.id, userId: localStorage.userId
            }
            this.$http.post('/library', params)
            this.product state = 'Добавлено'
        }
    },
    data() {
        return {
            product: null,
            product_state: 'Добавить'
        }
    },
    created() {
        const id = this.$route.params.id
        this.$http.get('/products/' + id)
            .then(response => this.product = response.data)
    }
}
</script>

```

## MainPage

```
<script>
export default {
  name: "main",
  data() {
    return{
      products: []
    }
  },
  created() {
    this.$http.get('/products')
      .then(response => this.products = response.data)
  }
}
</script>
```

## Результаты работы приложения:






AnimeGO

АнимеМангаО нас

Nicola Tesla



В библиотеке

Удалить

## Стальной алхимик: Братство

Fullmetal Alchemist: Brotherhood

Тип	TV Сериал
Эпизоды	64
Год выхода	2009
Студия	Bones
Режиссёр	Ириэ Ясуйиро
Возрастные ограничения	18+

Братья Эдварды отчаянно хотели воскресить погибшую маму, поэтому рискнули пренебречь непоколебимым правилом Алхимии, касающимся возрождения человека. В момент проведения запрещённого ритуала события начали разворачиваться по незапланированному сценарию. В результате финал оказался совершенно непредсказуемым. Альфонс полностью лишился плоти, а его душа стала пленницей железных доспехов. Эдвард также заплатил за свои деяния и лишился руки и ноги. Теперь парнишка вынужден пользоваться автоброней – специальным стальным обмундированием. Постепенно одарённый Эдвард превратился в государственного алхимика, став достойным представителем влиятельной военной машины своего государства. Но, несмотря на грандиозные достижения, персонаж не был счастлив, ведь он неустанно старался сделать так, чтобы его младший брат вернулся к прежний образ, а к нему его части тела. Служба в могущественной системе страны даёт родственникам большие привилегии, а также безграничный доступ к любой информации. Вскоре герои узнают о могущественном камне и отправляются на его поиски. Они даже не подозревают, какие испытания их ждут впереди.


СоглашениеКонфиденциальностьДля правообладателей

© animego.com 2021-2022

AnimeGO

АнимеМангаО нас

Nicola Tesla



Добавить

Удалить

## Ванпанчмен

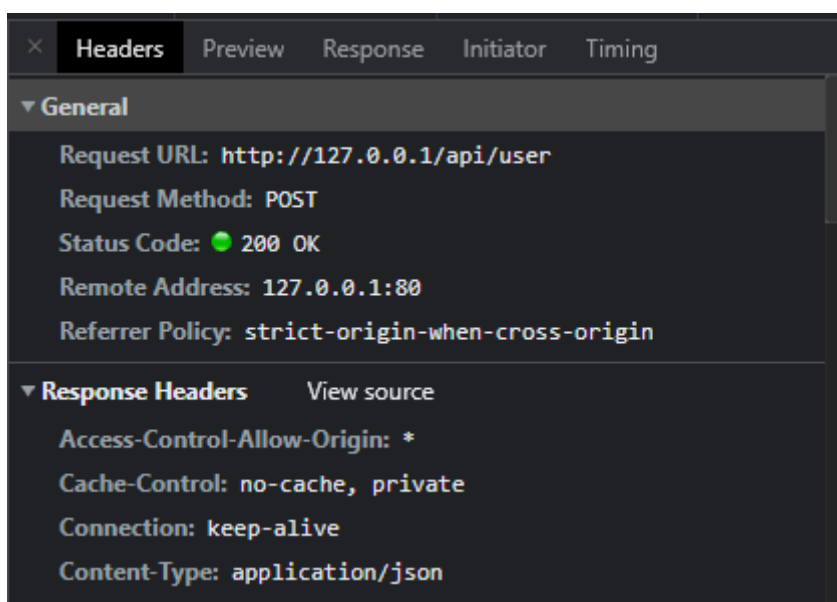
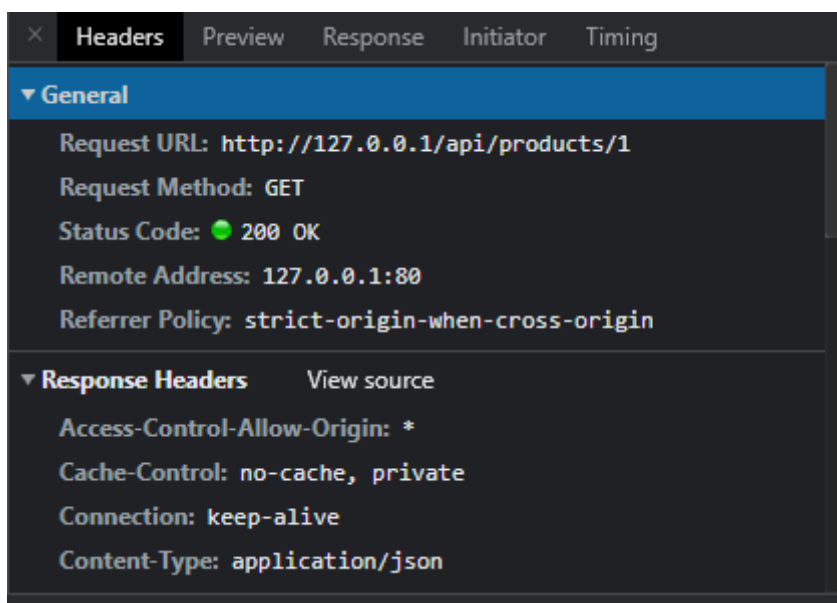
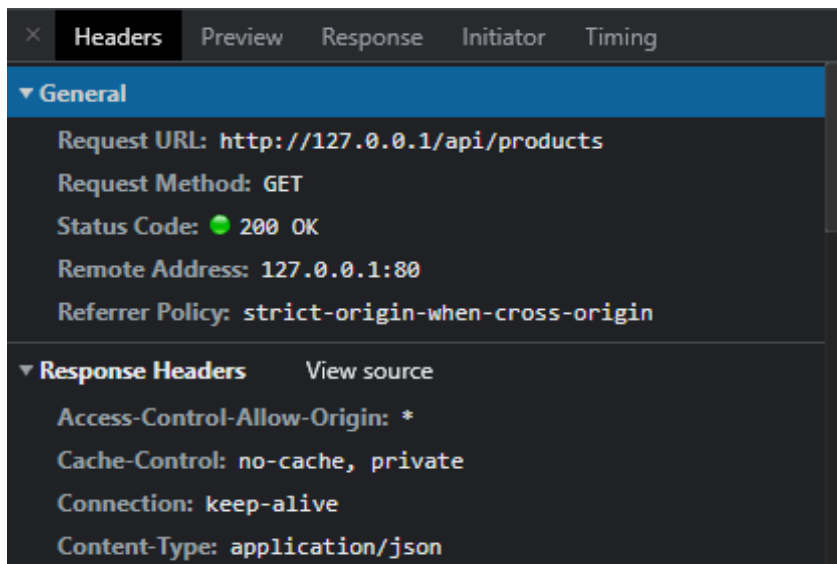
One Punch Man

Тип	TV Сериал
Эпизоды	12
Год выхода	2015
Студия	Madhouse Studios
Режиссёр	ONE
Возрастные ограничения	18+

Мы все привыкли, что комиксы наполнены различными злодеями, безумцами с планом захвата планеты и прочими неприятностями. Город Зет-Сити уже давно привык к такому роду событий, постоянными захватчиками из других миров и прочими бедствиями. Самое важное, чтобы всегда вовремя появился супергерой, который справится со всеми проблемами и сможет победить всех злодеев. Таким героем является парень по имени Сайтама. Вот только он выбивается из общего фона в жизни типичного города из комикса. Он невысокий, не может вести себя достаточно пафосно и не говорит высокопарных речей. Более того, внешний вид нашего героя состоит из простой одежды, а на голове нет ни одной волосинки. В прошлом Сайтама был обычным человеком, который потерял свою работу. Только благодаря своему терпению и постоянным тренировкам главный герой смог за три года стать самым сильным героем в мире. Чтобы победить врага, ему достаточно все лишь один раз ударить соперника, как тот окажется поверженным. Вот только теперь у него серьезная проблема – нет достойных соперников, и теперь вся это супергеройская жизнь превратилась для него в настоящую рутину.

СоглашениеКонфиденциальностьДля правообладателей

© animego.com 2021-2022



**Github:**

[https://github.com/NicoIaTesla/web\\_labs/tree/c0874f4643a02ac048ddf523d1ea148e9dee289f](https://github.com/NicoIaTesla/web_labs/tree/c0874f4643a02ac048ddf523d1ea148e9dee289f)