# 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

Технологии WEB-программирования «Работа с HTTP запросами»

Выполнила:

ст. группы ПВ-41

Бойко Валерия Евгеньевна

Проверил: Картамышев С. В.

**Цель работы:** изучить принципы работы отправки Ајах 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
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
 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('cart')->group(function(){
    Route::get('/{userId}', [\App\Http\Controllers\CartController::class, 'info']);
    Route::post('/', [\App\Http\Controllers\CartController::class, 'addToCart']);
});
Route::prefix('user')->group(function(){
    Route::post('/', [\App\Http\Controllers\UserController::class, 'authorization']);
});
```

#### Autorization.vue

```
<template>
  <section class = "authorization vue">
    <div class="login-box">
      <div class="row">
        <ima
            src="../assets/images/logo.png" alt="Logo"
        />
      </div>
      h2 \in \{ this.info \} < /h2 >
      <form>
        <div class="user-box">
          <input type="text" v-model="username" name="" required=""</pre>
class="username"/>
          <label><i class="Username"></i>
            Username
          </label>
        </div>
        <div class="user-box">
          <input type="password" v-model="password" name="" required=""</pre>
class="password"/>
          <label><i class="Unlock"></i>
            Password
          </label>
        </div>
        <button type="submit" class = "submit-button" @click = "authorization()"</pre>
>Bxog</button>
    </div>
  </section>
</template>
<script>
export default {
 name: "Authorization",
 beforeCreate: function() {
    document.body.className = 'auth';
  },
    data() {
        return {
            userId: null,
            username: null,
            password: null,
            info: 'Gallerix',
            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('MainPage')
                this.info = "Успешный вход"
                localStorage.userId = this.userId;
```

```
}
            else
            {
                this.info = "Неверные данные, повторите попытку"
            }
            console.log(this.correct)
        }
    },
   mounted(){
        if (localStorage.userId) {
            this.userId = localStorage.userId;
        }
    }
}
</script>
<style scoped src="../assets/styles/style.css">
```

#### ContentPage.vue

```
<template>
<section class="content">
<Header></Header>
  <main class ="container" v-if = "product">
    <div class="product-title">
     <h1>{{ product.title }}</h1>
    </div>
    <div class = "info">
      <div class="product-image">
        <img :src="require(`../assets/images/pictures/${product.id}.jpg`)"</pre>
class="picture-img-cover" alt="Картина">
     </div>
      <div class ="product-description">
       <div id="year"><h4>Год создания:</h4> {{product.year}} </div>
       <div class="authors"><h4>Abtop:</h4> {{ product.author }}</div>
       <div class="genre"><h4>Жанр живописи:</h4> {{ product.genre }}</div>
        <div class = "buying">
          <div class = "buying-price">
            <div class="buying-pricenew-text"><h4>Цена:</h4> {{product.price}}$</div>
          </div>
          <button class="btn" @click = "addToCart(product)"><span</pre>
class="text">Добавить <span>в корзину</span></span></button>
         </div>
     </div>
      <div class = "product-about">
       <h4>История создания картины:</h4>
       <h5>Автор: {{product.author}}<br>Название: "{{ product.title }}"</h5>
        <div class = "annotation">
          >
           {{product.description}}
          </div>
     </div>
    </div>
  </main>
  <Footer></Footer>
</section>
</template>
```

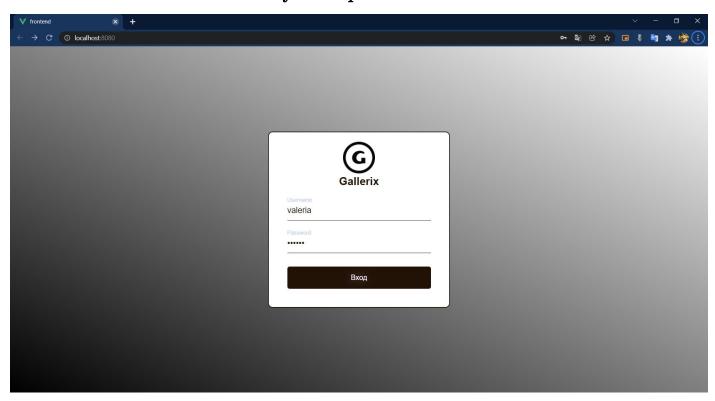
```
<script>
import Footer from "./Footer";
import Header from "./Header";
import Product from "./Product";
export default {
 name: "ContentPage",
  components: {
    Footer,
   Header,
    Product
  },
   methods: {
      addToCart(product) {
          const params = {
              id: product.id, count: 1
          this.$http.post('/cart', params)
      }
    },
  data() {
   return {
     product: null
    }
 },
 created() {
    const id = this.$route.params.id
    this. $http.get('/products/' + id)
      .then(response => this.product = response.data)
 }
}
</script>
<style scoped src="../assets/styles/content-style.css">
</style>
```

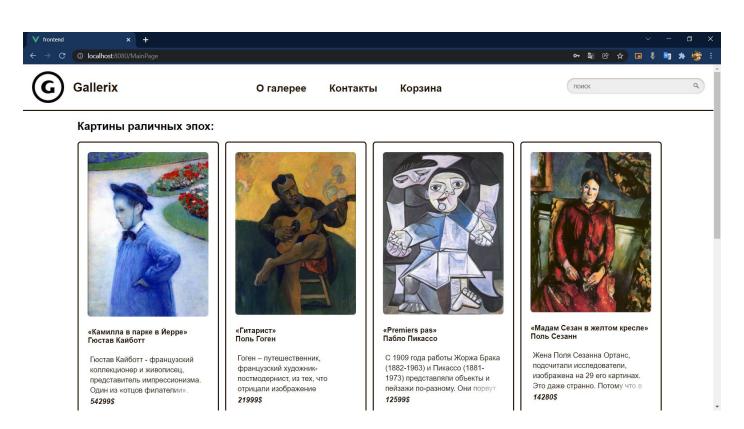
## MainPage.vue

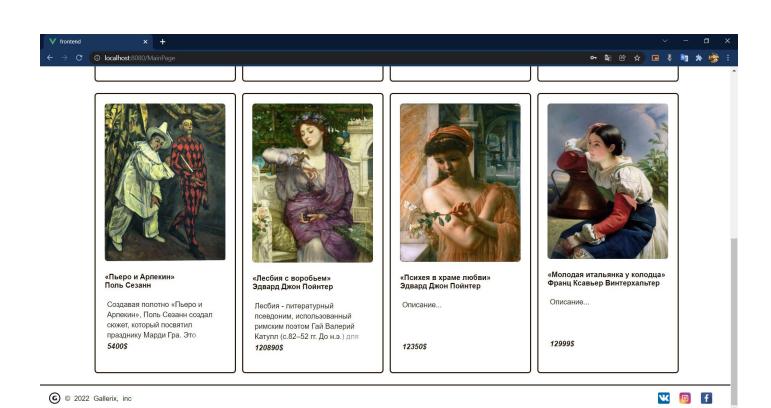
```
<template>
  <Header v-on:searchChanged = "updateSearch"></Header>
    <main class="container">
    <div>
        Картины раличных эпох:
      </h1>
    </div>
    <section class="products">
      <div class="v-for" v-for="product in products">
        <Product :product="product"></Product>
      </div>
    </section>
    </main>
  <Footer></Footer>
  </section>
</template>
<script>
import Product from "./Product";
import Header from "./Header";
import Footer from "./Footer";
import {forEach} from "core-js/internals/array-iteration";
export default {
 name: "MainPage",
```

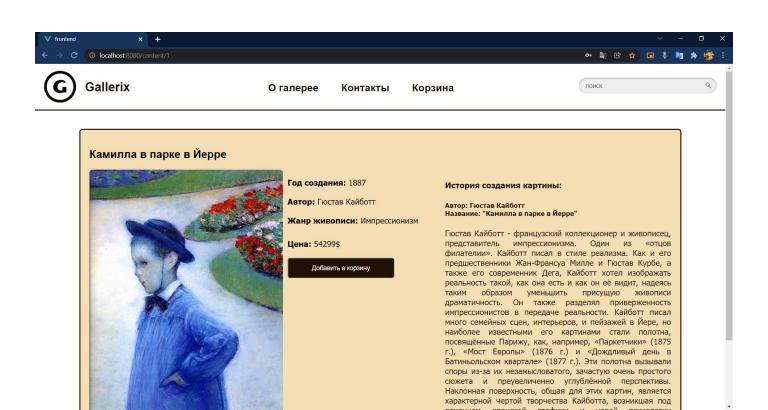
```
components: {
   Footer,
   Header,
   Product
 beforeCreate: function() {
   document.body.className = 'main';
 },
 data(){
   return{
     products: [],
     searched: "",
     allProducts: []
   }
  },
   created() {
       this.$http.get('/products')
           .then(response => this.searchFilter(response.data))
    },
   methods: {
      searchFilter(data){
              this.allProducts = data
              this.products = []
              data.forEach((element) => {
                  if (element.title.includes(this.searched)) {
                     this.products.push(element)
                 }
             })
     },
      updateSearch(e){
         this.searched = e
         this.searchFilter(this.allProducts)
     }
   }
}
</script>
<style scoped src="../assets/styles/main_style.css">
</style>
```

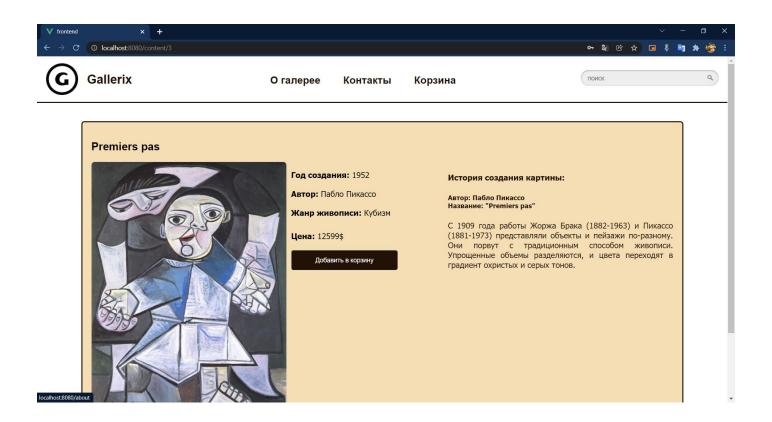
# Результат работы сайта













#### О галерее:

Gallerix.ru – частный некоммерческий проект, целью создания которого является популяризация искусства живописи среди широких масс населения и удовлетворения тяги к прекрасному у людей, не имеющих возможности лично посетить музеи, часто находящиеся на внушительном расстоянии от их дома.

Gallerix.ru – крупнейший виртуальный музей в русскоязычном интернете и один из крупнейших в мире, самый цитируемый сайт о живописи по оценке Яндекса.

Все картины, в обработке, представленной на сайте, олицетворяют личное мнение редакции о том, какими они могли быть в момент своего рождения. Мы обладаем только опубликованными файлами, ни в каком виде не высылаем ни отдельные изображения, ни архивы. Если вы хотите помочь в наполнении сайта или внести изменения в атрибуцию, изучите, пожалуйста, имеющиеся для этого возможности.

Сайт зарегистрирован Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций РФ, Свидетельство СМИ № ФС77-45143 от 19 мая 2011 г.

© © 2022 Gallerix, inc

ocalhost:8080/about





