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

Друздев Андрей Станиславович

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 программы:

authorization

<template>  
 <body>  
 <div class="login-box">  
 <div class="row">  
 <img  
 src="../assets/images/auto.png" alt="auto"  
 />  
 </div>  
 <h2>{{ this.info }}</h2>  
 <form>  
 <div class="user-box">  
 <input type="text" v-model="username" name="" required=""/>  
 <label><i class="Username"></i>  
 Почта или логин  
 </label>  
 </div>  
 <div class="user-box">  
 <input type="password" v-model="password" name="" required=""/>  
 <label><i class="Unlock"></i>  
 Пароль  
 </label>  
 </div>  
 </form>  
 <button type="submit" class="submit-button" @click = authorization()>Вход</button>  
 </div>  
  
 </body>  
</template>  
  
<script>  
export default {  
 name: "authorization",  
 data() {  
 return {  
 userId: null,  
 username: null,  
 password: null,  
 info: 'Авторизация',  
 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('main')  
 this.info = "Успешная авторизация"  
 ***localStorage***.userId = this.userId;  
 }  
 else  
 {  
 this.info = "Ошибка авторизации, повторите попытку"  
 }  
 ***console***.log(this.correct)  
 }  
 },  
 mounted(){  
 if (***localStorage***.userId) {  
 this.userId = ***localStorage***.userId;  
 }  
 }  
}  
</script>  
  
<style scoped src="../assets/styles/authorization\_style.css">  
  
</style>

Cart

<template>  
 <div class="cart">  
 <span v-if="cart">{{cart.total}} ₽</span>  
 </div>  
</template>  
  
<script>  
export default {  
 name: "cart",  
 data(){  
 return{  
 cart: null  
 }  
 },  
 created(){  
 this.$http.get('/cart/' + ***localStorage***.userId)  
 .then(response => this.cart = response.data)  
 }  
}  
</script>  
  
<style scoped>  
  
</style>

Footer

<template>  
 <footer class="footer">  
 <div class="footer-text">  
 © 2021-2022 Автозапчасти. All rights reserved.  
 </div>  
  
 <div class="footer-content-right">  
  
 <a href="https://vk.com/feed"><img src="../assets/images/vk.png" class="icon-style" alt="vk group"></a>  
  
 </div>  
 </footer>  
</template>  
  
<script>  
export default {  
 name: "footer"  
}  
</script>  
  
<style scoped src="../assets/styles/footer\_style.css">  
  
</style>

Header

<template>  
 <header class = main-header>  
  
 <div class="header-text">  
 <h1>Автозапчасти</h1>  
 </div>  
  
 <img src = "../assets/images/auto.png" alt ="auto" />  
  
  
 <nav><a href="https://emex.ru/"> <h2>О нас</h2> </a> <a href="#"><h2>Корзина<app-cart></app-cart></h2> </a></nav>  
  
 </header>  
</template>  
  
<script>  
export default {  
 name: "header"  
}  
</script>  
  
<style scoped src="../assets/styles/header\_style.css">  
  
</style>

Main

<template>  
 <section>  
 <app-header></app-header>  
  
 <main class="container">  
 <div>  
 <h1>  
 Хиты продаж  
 </h1>  
 </div>  
 <section class="products">  
 <div class="v-for" v-for="product in products" :key="product.id">  
 <app-product :product="product"> </app-product>  
 </div>  
 </section>  
 </main>  
  
 <app-footer></app-footer>  
 </section>  
  
</template>  
  
<script>  
export default {  
 name: "main",  
 data(){  
 return{  
 products: []  
 }  
 },  
 created() {  
 this.$http.get('/products')  
 .then(response => this.products = response.data)  
 }  
}  
</script>  
  
<style scoped src="../assets/styles/main\_style.css">  
  
</style>

Product

<template>  
 <section class="product-item">  
 <div>  
 <img :src=product.img alt="">  
 </div>  
 <h4>  
 <div class="title'">  
 <router-link :to="{name: 'product\_page', params: {product: product, id: product.id}}">  
 {{ product.title }}  
 </router-link>  
 </div>  
 </h4>  
 <p>  
 {{product.description}}  
 </p>  
 <div>  
 <span class="price">{{product.price}} ₽  
 </span>  
 </div>  
 </section>  
</template>  
  
<script>  
export default {  
 name: "product",  
 props: ['product']  
}  
</script>  
  
<style scoped src="../assets/styles/product\_style.css">  
  
</style>

Product\_page

<template>  
 <section>  
 <app-header></app-header>  
 <main class="container">  
 <div class="product-title">  
 <h1>{{product.title}}</h1>  
 </div>  
  
 <div class="info">  
 <div class="product-image">  
 <img :src= product.img alt="">  
 </div>  
  
 <div class="product-description">  
 <div class="manufacturer">  
 Производитель: {{product.manufacturer}}  
 </div>  
  
 <div class="buying">  
 <div class="buying-price">  
 <div>Цена:</div>  
 <div class="buying-pricenew-val">  
 {{product.price}} ₽  
 </div>  
 </div>  
 <a @click = "addToCart(product)" class="btn"><span class="text">Добавить в корзину</span></a>  
  
 </div>  
 </div>  
  
 <div class="product-about">  
 <h4>Инормация о товаре</h4>  
 <div>  
 {{product.description}}  
 </div>  
 </div>  
 </div>  
  
  
 </main>  
 <app-footer></app-footer>  
 </section>  
  
</template>  
  
<script>  
  
export default {  
 name: "product\_page",  
 methods: {  
 addToCart(product){  
 const params = {  
 id: product.id, count: 1, userId: ***localStorage***.userId  
 }  
 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/product\_page\_style.css">  
  
</style>

Api <?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('user')->group(function(){  
 Route::post('/', [\App\Http\Controllers\UserController::class, 'authorization']);  
});  
  
Route::prefix('cart')->group(function(){  
 Route::post('/', [\App\Http\Controllers\CartController::class, 'addToCart']);  
 Route::get('/{userId}', [\App\Http\Controllers\CartController::class, 'info']);  
});  
  
Route::prefix('products')->group(function(){  
 Route::get('/', [\app\Http\Controllers\ProductController::class, 'list']);  
 Route::get('/{id}', [\App\Http\Controllers\ProductController::class, 'info']);  
});

Скриншоты выполнения программы:

 

