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

Технологии Web-программирования

Лабораторная работа №3

**Серверное программирование. Yii2**

Выполнил: студент группы ВТ-41

Горохов В.О.

Проверил:

Картамышев С.В.

Белгород 2020

**Цель:** познакомиться с основами backend разработки web-приложений. Научится писать скрипты на языке PHP. Познакомиться с основами работы docker. Познакомиться с фреймворком Yii2 и научиться разворачивать проект, производить его настройку. Научится работать с API в приложении Postman.

**Задание к лабораторной работе:**

1. Развернуть базовое приложение Yii2 App Basic.
2. Настроить конфигурацию работы приложения с docker.
3. Добавить модуль для работы с API.
4. Добавить несколько контроллеров со статическими данными.
5. Продемонстрировать работу API в Postman.

**Выполнение**

Для работы был установлен менеджер пакетов composer, который отвечает за установку и взаимодействие библиотек. (В js был npm)

После установки менеджера пакетов была выполнена команда, которая разворачивает базовое приложение:

composer create-project --prefer-dist yiisoft/yii2-app-basic basic

В существующем проекте создалась папка basic, ее содержимое было перенесено в корень проекта.

На localhost открылась страница приветствия.

**При последующих запусках проекта просто понимались пакеты командой:**

docker-compose up -d или через приложение docker.

Далее были настроены конфигурации в директории config, файлы db.php и web.php.

В файл db.php были записаны данные из файла docker-compose.yml.

class' => 'yii\db\Connection',

'dsn' => 'mysql:host=mysql;dbname=bstu',

'username' => 'bstu',

'password' => 'bstu321!',

'charset' => 'utf8',

В web.php раскомментировал urlManager для корректного отображения ссылок.

**Основные составляющие Фреймворка YII:**

Компонент model - работает с данными.

Компонент vue - работает с представлениями.

Контроллер - связывает модель и представление.

Контроллер обращается к модели, модель к БД. Затем контроллер отдает эту модель представлению. Представление получает данные в воде модели и отображает их;

### Создание API

API - интерфейс взаимодействия клиента и сервера.

Создал директорию modules, в ней создал директорию v1

v1 - версия 1. Когда создают API, указывают версионность. Это необходимо для того, что бы не сломать все приложения, которые работали по другой (старой) версии.

В директории v1 создал php класс v1 и унаследовал его от класса module. Далее подключил его в конфигурациях.

После подключения к конфигурациям создал страницы.

Для этого в директории v1 создал директорию controllers и добавил в нее класс контроллер ...Controller.php, который будет наследоваться от rest\Controller (т.к. создаем api) затем в нем описал метод action...()

В дальнейшем **для тестирования API** используют сторонние сервисы, один из них это **postman**

<https://web.postman.co/home> - CreateNew - Create a request

В postman произошла cors ошибка. Установил desktop agent и отправил запрос еще раз. Заработало. Эта ошибка говорит о том, что нельзя выполнять кросс доменные запросы c разных хостов и портов. Это сделано для того, что бы один сайт не мог выполнять запросы к другим сервисам не связанным с текущим хостом.

Иным решение этой ошибки является набор заголовков, который возвращаться вместе с данными при запросе. Этот набор разрешает кросс доменные запросы.

Для этого из класса Controller, rest контроллеров, переопределил метод behaviors в своем классе ApiController. Удалил из этого метода поле xlm и добавил элемент, который и будет отправлять заголовки:

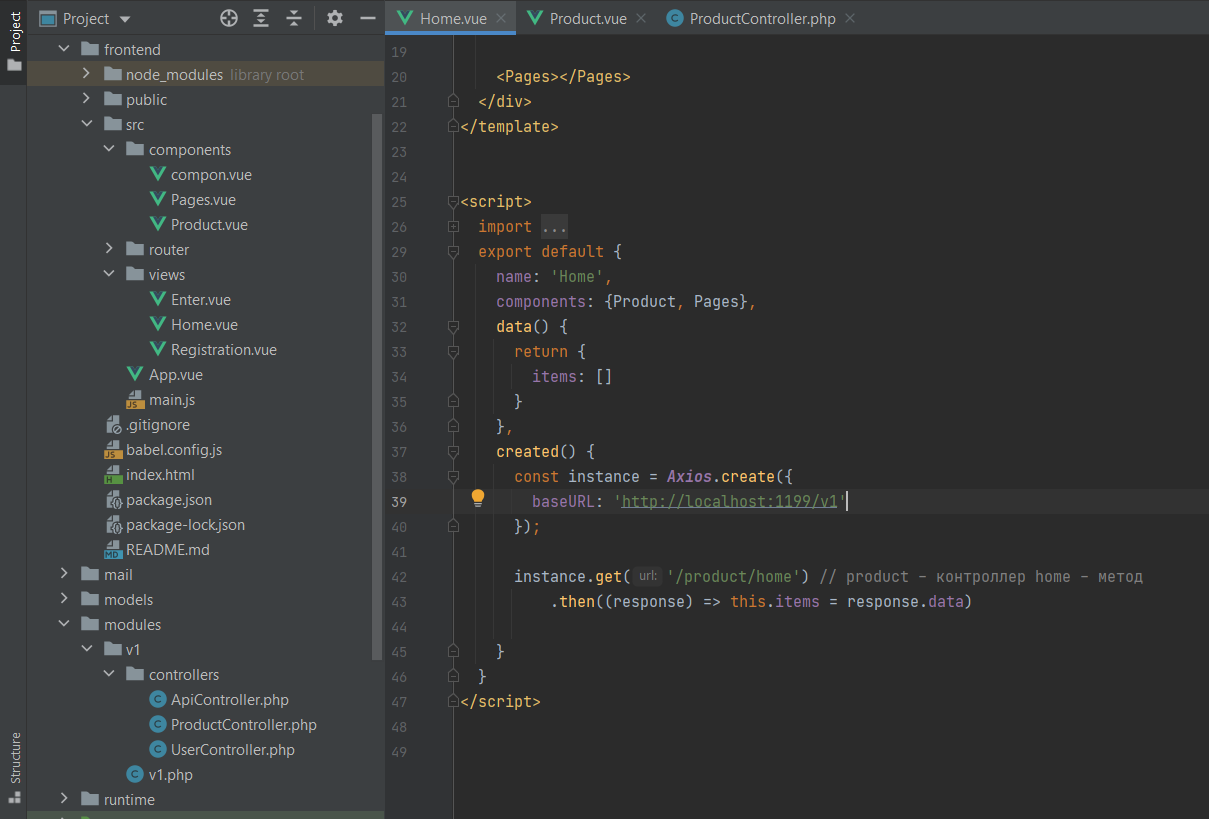
'corsFilter' => [  
   'class' => Cors::class  
]

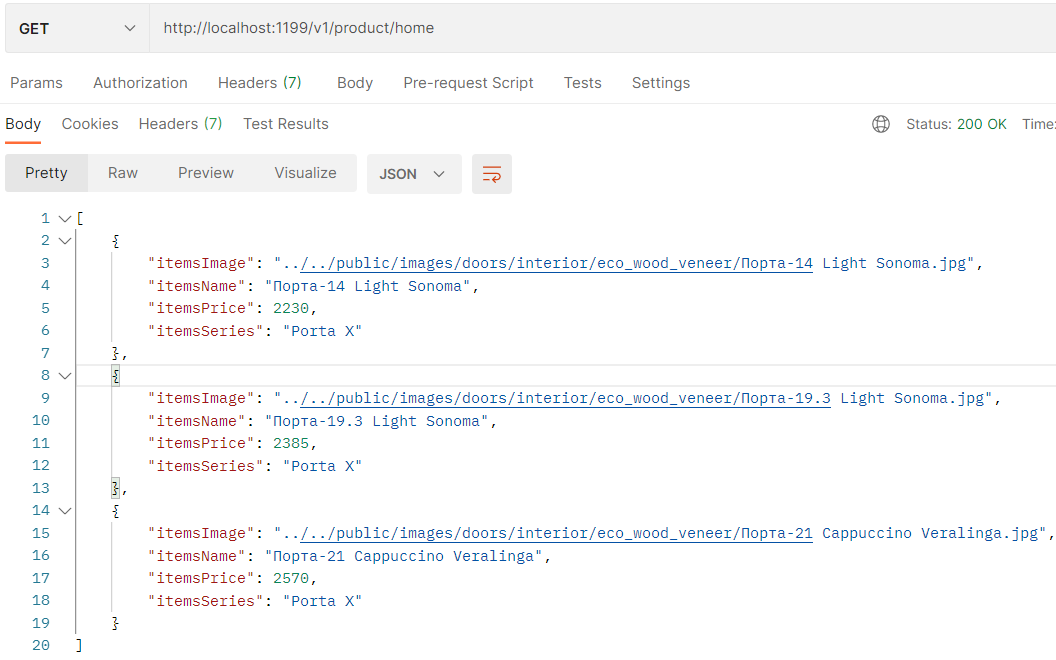
Далее, ранее созданный класс из директории controllers, унаследовал не от rest контроллера, а от API контроллера.

Создал статический контроллер товаров ProductController и подключил его к frontend предварительно выполнив npm run serve

**Для связки vue и контроллеров использовал библиотеку axios, предварительно установив ее командой:** npm install axios

Протестировал api в **postman**





Приложение 1. Основные исходные коды.

«ApiController.php»:

<?php  
  
namespace app\modules\v1\controllers;  
  
use yii\filters\auth\CompositeAuth;  
use yii\filters\ContentNegotiator;  
use yii\filters\Cors;  
use yii\filters\RateLimiter;  
use yii\filters\VerbFilter;  
use yii\rest\Controller;  
use yii\web\Response;  
  
class ApiController extends Controller {  
 public function behaviors () {  
 return [  
 'contentNegotiator' => [  
 'class' => ContentNegotiator::className(),  
 'formats' => [  
 'application/json' => Response::FORMAT\_JSON,  
 ],  
 ],  
 'verbFilter' => [  
 'class' => VerbFilter::className(),  
 'actions' => $this->verbs(),  
 ],  
 'authenticator' => [  
 'class' => CompositeAuth::className(),  
 ],  
 'rateLimiter' => [  
 'class' => RateLimiter::className(),  
 ],  
 'corsFilter' => [  
 'class' => Cors::class  
 ]  
 ];  
 }  
}

«ProductController.php»:

<?php  
namespace app\modules\v1\controllers;  
  
class ProductController extends ApiController {  
 public function actionHome () {  
 $products = [  
 [  
 'itemsImage' => '../../public/images/doors/interior/eco\_wood\_veneer/Порта-14 Light Sonoma.jpg',  
 'itemsName' => 'Порта-14 Light Sonoma',  
 'itemsPrice' => 2230,  
 'itemsSeries' => 'Porta X'  
 ],  
 [  
 'itemsImage' => '../../public/images/doors/interior/eco\_wood\_veneer/Порта-19.3 Light Sonoma.jpg',  
 'itemsName' => 'Порта-19.3 Light Sonoma',  
 'itemsPrice' => 2385,  
 'itemsSeries' =>'Porta X'  
 ],  
 [  
 'itemsImage' => '../../public/images/doors/interior/eco\_wood\_veneer/Порта-21 Cappuccino Veralinga.jpg',  
 'itemsName' => 'Порта-21 Cappuccino Veralinga',  
 'itemsPrice' => 2570,  
 'itemsSeries' => 'Porta X'  
 ]  
 ];  
 return $products;  
 }  
}

« UserController.php»:

<?php

namespace app\modules\v1\controllers;

class UserController extends ApiController {

public function actionIndex() {

$users = [

[

'id' => 1,

'name' => 'Ivan',

'age' => 23

],

[

'id' => 2,

'name' => 'Vasia',

'age' => 45

]

];

return $users;

}

}

«Home.vue»:

<!-- Каждый элемент во vue представляется в виде компонента. -->  
<!-- Каждый компонент состоит из 3-х секций: template (html шаблон), script (js код исполняемый на стр.) -->  
<!-- и style секция для специфичных стилей, которые используются внутри template (вместо style лучше создать css файл). -->  
  
<template>  
 <div class="home">  
 <div class="container">  
 <h2 align="center"> Двери в Белгороде с покрытием экошпона! </h2>  
 <h3 align="center"> Дополнительную информацию по товару вы можете уточнить у менеджеров магазина. </h3>  
 </div>  
  
 <div class="container">  
 <div class="divider margin\_bottom\_50">  
 <div v-for="item in items">  
 <Product v-bind:productParam="item"> </Product>  
 </div>  
 </div>  
 </div>  
  
 <Pages></Pages>  
 </div>  
</template>  
  
  
<script>  
 import Axios from 'axios';  
 import Product from "@/components/Product";  
 import Pages from "@/components/Pages";  
 export default {  
 name: 'Home',  
 components: {Product, Pages},  
 data() {  
 return {  
 items: []  
 }  
 },  
 created() {  
 const instance = Axios.create({  
 baseURL: 'http://localhost:1199/v1'  
 });  
  
 instance.get('/product/home') // product - контроллер home - метод  
 .then((response) => this.items = response.data)  
  
 }  
 }  
</script>

«Product.vue»:

<template>

<div class="product">

<div class="container\_doors">

<div class="img\_and\_text">

<!-- <img v-bind:src="productParam.itemsImage" height="300"> <br>-->

<img src="../../public/images/doors/interior/eco\_wood\_veneer/Порта-14 Light Sonoma.jpg" height="300"> <br>

{{productParam.itemsName}} <br>

<b> {{ productParam.itemsPrice }} руб. </b>

</div>

<div class="door\_info">

<div class="container\_local">

<div class="divider">

<div>

<img src="../../public/images/doors/interior/eco\_wood\_veneer/Порта-14 Light Sonoma.jpg" height="300"> <br>

</div>

<div class="margin\_left\_30">

<b> Серия: </b> {{productParam.itemsSeries}} <br>

<b> Размеры: </b> <br>

200\*60 <br> 200\*70 <br> 200\*80 <br> 200\*90 <br>

</div>

</div>

</div>

</div>

</div>

</div>

</template>

<script>

export default {

name: "Product",

// props принемает на вх. данные из Home.vue items

props: ['productParam']

}

</script>