**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**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 №3  
дисциплина: Технологии Web-программирования  
тема: «Серверное программирование. Yii2»

Выполнила: ст. группы ПВ-41  
Немшилова А.Г.  
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 работы:**

Развернули базовое приложение Yii2 App Basic используя команду composer create-project --prefer-dist yiisoft/yii2-app-basic basic.

Для настройки конфигурации изменили файл docker-compose.yaml:

version: '3.1'  
services:  
 php:  
 image: yiisoftware/yii2-php:7.3-apache  
 volumes:  
 - ~/.composer-docker/cache:/root/.composer/cache:delegated  
 - ./:/app:delegated  
 ports:  
 - '1199:80'  
  
 mysql:  
 image: mysql  
 command: --default-authentication-plugin=mysql\_native\_password --character-set-server=utf8 --collation-server=utf8\_general\_ci  
 environment:  
 - MYSQL\_ROOT\_PASSWORD=verysecret  
 - MYSQL\_DATABASE=bstu  
 - MYSQL\_USER=bstu  
 - MYSQL\_PASSWORD=bstu321!

Добавили следующий модуль и некоторые контроллеры для работы со статическими данными:

**User.php**

<?*php*namespace app\modules\v1\models;  
use yii\base\Model;  
  
class User extends Model {  
  
 public function login() {  
 // do login  
 }  
  
 public function register() {  
 // do register  
 }  
  
 public function getUsers() {  
 return [  
 [  
 'id' => *1*,  
 'login' => "login1",  
 'password' => 'pass1',  
 ],  
 [  
 'id' => *2*,  
 'login' => "login2",  
 'password' => 'pass2',  
 ],  
 [  
 'id' => *3*,  
 'login' => "login3",  
 'password' => 'pass3'  
 ],  
 ];  
 }  
}

**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  
 ]  
 ];  
 }  
}

<?*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  
 ]  
 ];  
 }  
}

**UserController.php:**

<?*php*namespace app\modules\v1\controllers;  
  
use app\modules\v1\models\User;  
use yii\rest\Controller;  
  
class UserController extends ApiController {  
 public function actionLogin() {  
 // login stuff  
 }  
  
 public function actionRegister() {  
 // register stuff  
 }  
  
 public function actionAll() {  
 $user = new User();  
 return $user -> getUsers();  
 }  
}

**ProductsController.php:**

<?*php*namespace app\modules\v1\controllers;  
  
class ProductsController extends ApiController {  
 public function actionCatalog() {  
 $products = [  
 [  
 'id' => *1*,  
 'name' => 'Торт "Секрет Небес"',  
 'url' => 'Секрет Небес',  
 'image' => 'https://madbutterbd.com/images/mad\_butter/products/1/image.jpg',  
 'price' => *3500*,  
 ],  
 [  
 'id' => *2*,  
 'name' => 'Капкейки "Единорог"',  
 'url' => 'Единорог',  
 'image' => 'https://i.pinimg.com/originals/69/42/b8/6942b8c6c23c9705e82055a5f27e57ce.jpg',  
 'price' => *650*,  
 ],  
 [  
 'id' => *3*,  
 'name' => 'Торт "Космос"',  
 'url' => 'Единорог',  
 'image' => 'https://i.pinimg.com/736x/92/79/f7/9279f78aa0e02295cb9d31d1464dbb38.jpg',  
 'price' => *3500*,  
 ]  
 ];  
  
 return $products;  
 }  
}

Результаты работы запросов в Postman:



