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

Клесов М.И.

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.

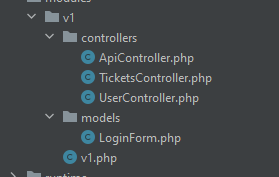
**Задание к лабораторной работе:**

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.

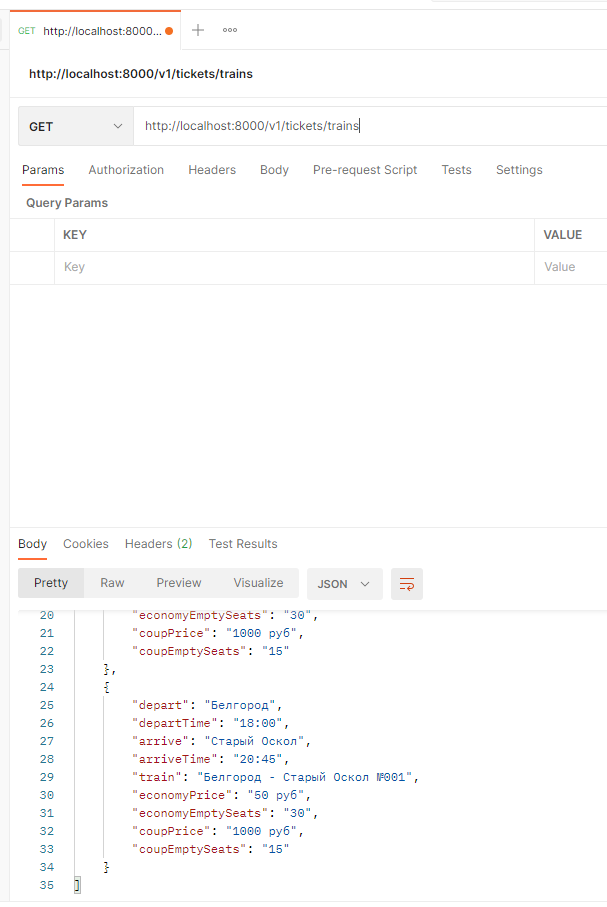
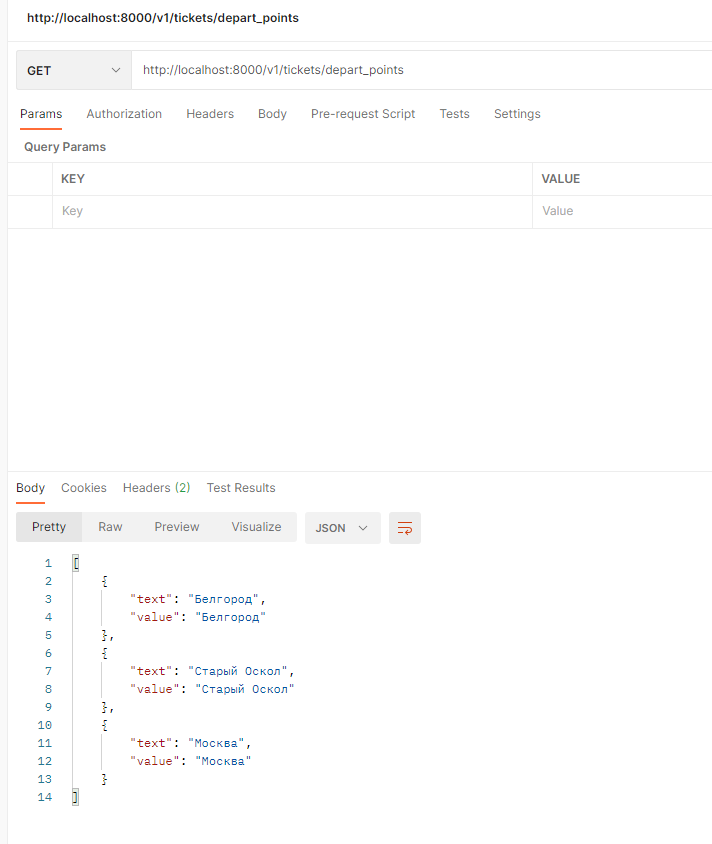
**Выполнение**

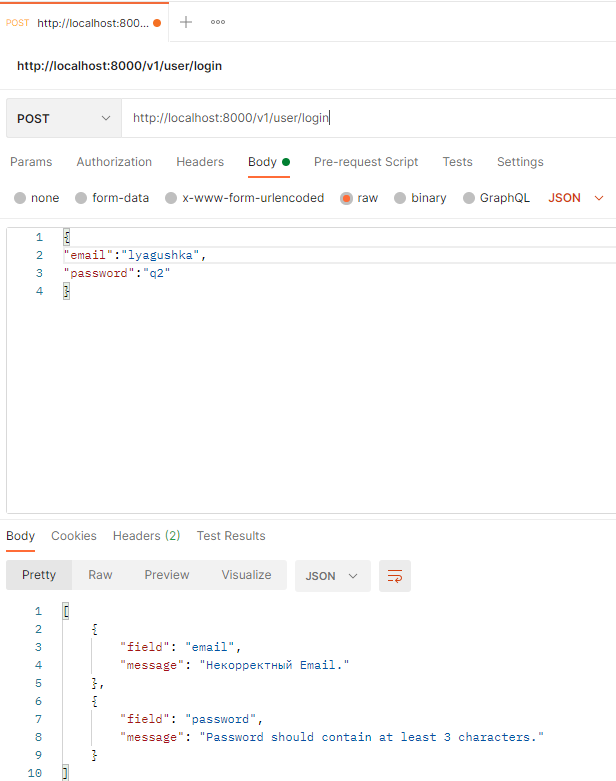
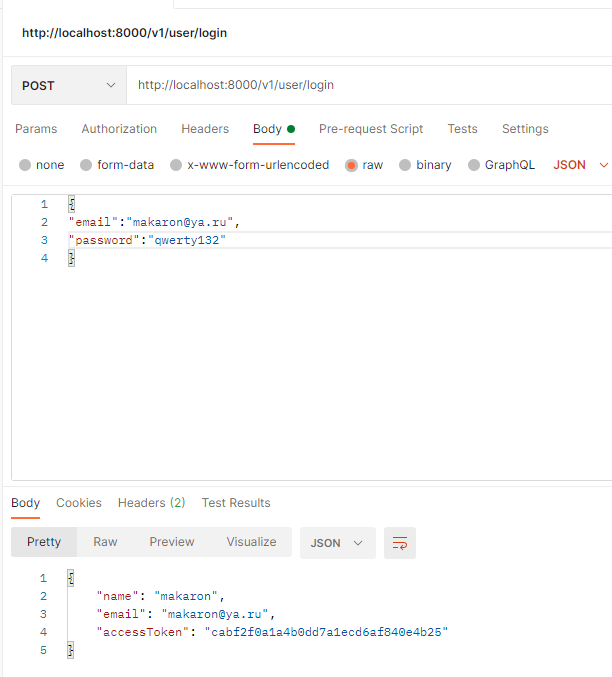
В ходе работы было развёрнуто базовое приложение Yii2 App Basic с помощью команды composer create-project --prefer-dist yiisoft/yii2-app-basic basic. Затем это приложение было настроено для работы и дополнено результатами предыдущих лабораторных работ. Константные данный были перенесены в бэкенд сайта и вызываются посредством REST API.

Созданные компоненты серверной части:



Тестирование 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//,  
 //'application/xml' => Response::FORMAT\_XML,* ],  
 ],  
 'verbFilter' => [  
 'class' => VerbFilter::*className*(),  
 'actions' => $this->verbs(),  
 ],  
 'authenticator' => [  
 'class' => CompositeAuth::*className*(),  
 ],  
 'rateLimiter' => [  
 'class' => RateLimiter::*className*(),  
 ],  
 'corsFilter' => [  
 'class' => Cors::class  
 ],  
 ];  
 }  
}

«TicketController.php»:

<?php  
  
  
namespace app\modules\v1\controllers;  
  
  
class TicketsController extends ApiController {  
 public function actionDepart\_points() {  
 return [  
 [ 'text' => 'Белгород', 'value' => 'Белгород' ],  
 [ 'text' => 'Старый Оскол', 'value' => 'Старый Оскол' ],  
 [ 'text' => 'Москва', 'value' => 'Москва' ]  
 ];  
 }  
 public function actionArrive\_points() {  
 return [  
 [ 'text' => 'Белгород', 'value' => 'Белгород' ],  
 [ 'text' => 'Старый Оскол', 'value' => 'Старый Оскол' ],  
 [ 'text' => 'Москва', 'value' => 'Москва' ]  
 ];  
 }  
 public function actionTrains() {  
 return [  
 [ 'depart' => 'Белгород',  
 'departTime'=> '8:00',  
 'arrive'=> 'Старый Оскол',  
 'arriveTime' => '10:45',  
 'train' => 'Белгород - Старый Оскол №001',  
 'economyPrice' => '50 руб',  
 'economyEmptySeats' => '30',  
 'coupPrice' => '1000 руб',  
 'coupEmptySeats' => '15' ],  
  
 [ 'depart' => 'Белгород',  
 'departTime'=> '11:00',  
 'arrive'=> 'Старый Оскол',  
 'arriveTime' => '13:45',  
 'train' => 'Белгород - Старый Оскол №001',  
 'economyPrice' => '50 руб',  
 'economyEmptySeats' => '30',  
 'coupPrice' => '1000 руб',  
 'coupEmptySeats' => '15' ],  
  
 [ 'depart' => 'Белгород',  
 'departTime'=> '18:00',  
 'arrive'=> 'Старый Оскол',  
 'arriveTime' => '20:45',  
 'train' => 'Белгород - Старый Оскол №001',  
 'economyPrice' => '50 руб',  
 'economyEmptySeats' => '30',  
 'coupPrice' => '1000 руб',  
 'coupEmptySeats' => '15' ]  
 ];  
 }  
}

«UserController.php»:

<?php  
  
namespace app\modules\v1\controllers;  
use app\modules\v1\models\LoginForm;  
  
class UserController extends ApiController {  
 public function actionLogin() {  
 $form = new LoginForm();  
 $form->load(\Yii::*$app*->request->getBodyParams(), '');  
  
 if ($form->validate()) {  
 $form->login();  
 return $form->getUser();  
 }  
 *//если в форме есть ошибка, то возвращает 422 код и список ошибок* return $form;  
 }  
}

«LoginForm.php»:

<?php  
  
namespace app\modules\v1\models;  
  
use yii\base\Model;  
  
class LoginForm extends Model {  
 public $email;  
 public $password;  
  
 public function rules() {  
 return [  
 [['email', 'password'], 'required'],  
 [['email'], 'email', 'message' => 'Некорректный Email.'],  
 [['email', 'password'], 'string', 'max' => 64, 'min' => 3]  
 ];  
 }  
  
 public function login() {  
 *// TODO: сделать авторизацию* }  
  
 public function getUser() {  
 $str = *strpos*($this->email, "@");  
 return [  
 'name' => *substr*($this->email, 0, $str),  
 'email' => $this->email,  
 'accessToken' => *md5*(*microtime*(null))  
 ];  
 }  
}

«Home.vue»:

<template>  
 <div class="bg\_gray">  
 <h1 class="text-center form-header"> ЖД билеты онлайн</h1>  
 <form class="form-travel">  
 <div class="form-row row-travel">  
 <div class="col-md-3 mb-3">  
 <label for="select-from">Откуда</label>  
 <select class="form-control" id="select-from" v-model="depart">  
 <option disabled value="">Выберите пункт отправления</option>  
 <option v-for="option in depart\_options" v-bind:value="option.value">  
 {{ option.text }}  
 </option>  
 </select>  
 </div>  
 <div class="col-md-3 mb-3">  
 <label for="select-to">Куда</label>  
 <select class="form-control" id="select-to" placeholder="Не выбрано" v-model="arrive">  
 <option disabled value="">Выберите пункт прибытия</option>  
 <option v-for="option in arrive\_options" v-bind:value="option.value">  
 {{ option.text }}  
 </option>  
 </select>  
 </div>  
 <div class="col-md-3 mb-3">  
 <label for="travelDate">Дата</label>  
 <input class="form-control" type="date" id="travelDate" v-model="date">  
 </div>  
 <div class="col-md-3 mb-3">  
 <router-link class="btn btn-danger btn-block btn-bot"  
 v-bind:to="{name: 'RoutsTable', params: {depart: depart, arrive : arrive, date: date}}">Найти билеты</router-link>  
*<!-- <button class="btn btn-danger btn-block btn-bot" v-on:click="btnClick">Найти билеты</button>-->* </div>  
 </div>  
 </form>  
 </div>  
</template>  
  
<script>  
import Axios from 'axios';  
export default {  
 name: 'Home',  
 components: {},  
 data() {  
 return {  
 depart: "",  
 arrive: "",  
 date: "",  
 depart\_options: [],  
 arrive\_options: []  
 }  
 },  
 created() {  
 const instance = Axios.create({  
 baseURL: 'http://localhost:8000/v1'  
 });  
 instance.get('/tickets/depart\_points').then((response) => this.depart\_options = response.data)  
 instance.get('/tickets/arrive\_points').then((response) => this.arrive\_options = response.data)  
 },  
 methods: {  
 btnClick(){  
 }  
 }  
}  
</script>