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 №5

дисциплина: «Технологии Web-программирования»

тема: «REST API»

Выполнил: студент группы ВТ-41

Шкорба В. С.

Проверил: Картамышев С.В.

Белгород 2020

**Цель:** изучить основы разработки API для web-приложений. Разработать REST API для своего проекта.

**Задание к лабораторной работе:**

1. Изучить структуру формата представления данных JSON.
2. Изучить типы запросов к API: HEAD, GET, POST, PUT, DELETE.
3. Спроектировать и реализовать собственное REST API (Получение, создание, изменение и удаление каких-либо объектов).
4. В отчёт необходимо предоставить документацию к использованию методов. (Либо словесным описание, либо через Swagger)

**Выполнение**

В ходе выполнения работы было разработано REST API к проекту, а также дополнены функционал моделей БД.

**Общее описание к методам:**

Базовый URL: <http://192.168.0.195::1199/v1>

**Контроллеры:**

1. Контроллер категорий товаров CategoryController

URL: /category

Тип запроса: GET

Действия:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название/url | Описание | Параметры | | |
| параметр | тип | обязательный |
| actionList | Возвращает все категории | Нет |  |  |
| /list |

1. Контроллер заказов OrderController

URL: /order

Тип запроса: POST

Действия:

|  |  |  |
| --- | --- | --- |
| Название/url | Описание | Данные |
| actionCreate | Создаёт заказ с привязкой к пользователю и продукту по их id | {userId, productId, price} |
| /create |

Тип запроса: POST

Действия:

|  |  |  |
| --- | --- | --- |
| Название/url | Описание | Данные |
| actionDelete | Удаляет заказ по id пользователя и id продукта | {userId, productId, price} |
| /delete |

1. Контроллер страниц PageController

URL: /page

Тип запроса: GET

Действия:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название/url | Описание | Параметры | | |
| параметр | тип | обязательный |
| actionDelivery | Возвращает информацию о странице «Доставка» | Нет |  |  |
| /delivery |
| actionContact | Возвращает информацию о странице «Контакты» | Нет |  |  |
| /contact |

1. Контроллер товаров ProductController

URL: /product

Тип запроса: GET

Действия:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название/url | Описание | Параметры | | |
| параметр | тип | обязательный |
| actionHome | Возвращает 3 последних добавленных товара | Нет |  |  |
| /home |
| actionCatalog | Возвращает все товары | Нет |  |  |
| /catalog |
| actionInfo | Возвращает информацию о товаре | url | String(128) | Да |
| /info |

1. Контроллер пользователей UserController

URL: /user

Тип запроса: GET

Действия:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название/url | Описание | Параметры | | |
| параметр | тип | обязательный |
| actionLogin | Выполняет авторизацию пользователя по его email и password. Возвращает соответствующего пользователя или исключение с кодом 406 – неверный логин или пароль | email  password | String(100)  String(50) | Да  Да |
| /login |

Тип запроса: POST

Действия:

|  |  |  |
| --- | --- | --- |
| Название/url | Описание | Данные |
| actionRegistration | Создаёт пользователя с указанным email, name и password, если пользователя с таким email не существует, иначе – исключение с кодом 406 | {email, name, password} |
| /registration |

Пример реализации контроллера:

<?php

namespace app\modules\v1\controllers;

use app\modules\v1\models\Product;

use yii\web\NotFoundHttpException;

use Yii;

class ProductController extends ApiController

{

    public function actionHome()

    {

        $products = Product::find()

            ->orderBy(['createdAt' => SORT\_DESC])

            ->limit(3)

            ->all();

        return $products;

    }

    public function actionCatalog()

    {

        $products = Product::find()

            ->orderBy(['createdAt' => SORT\_DESC])

            ->all();

        return $products;

    }

    public function actionCreate()

    {

        $model = new Product();

        $model->load(Yii::$app->request->getBodyParams(), '');

        $model->save();

        return $model;

    }

    public function actionInfo($url)

    {

        $product = Product::find()

            ->with(['category', 'propertiesValues', 'propertiesValues.property'])

            ->where(['url' => $url])

            ->one();

        if ($product === null) {

            throw new NotFoundHttpException('Товар не найден');

        }

        return $product;

    }

}