***Пользовательская часть***

1. **Авторизация**

Производится посредством отправки POST запроса на относительный адрес “/login” с передачей в формате json следующих параметров:

1. email – email пользователя
2. password – пароль пользователя

***Пример вызова:***

$curl = curl\_init();

curl\_setopt\_array($curl, **array**(

  CURLOPT\_URL => 'http://exbico/public/login',

  CURLOPT\_RETURNTRANSFER => **true**,

  CURLOPT\_ENCODING => '',

  CURLOPT\_MAXREDIRS => 10,

  CURLOPT\_TIMEOUT => 0,

  CURLOPT\_FOLLOWLOCATION => **true**,

  CURLOPT\_HTTP\_VERSION => CURL\_HTTP\_VERSION\_1\_1,

  CURLOPT\_CUSTOMREQUEST => 'POST',

  CURLOPT\_POSTFIELDS =>'{

    "email" : "example@mail.ru",

    "password" : "exbico"

}',

  CURLOPT\_HTTPHEADER => **array**(

    'Content-Type: application/json'

  ),

));

$response = curl\_exec($curl);

curl\_close($curl);

**echo** $response;

1. **Запуск скрипта через REST API.**

Пройзводится посредством отправки POST запроса на относительный адрес “/calc” c передачей в формате json следующих параметров:

1. intParam – целое число
2. arParams – массив целых чисел

***Пример вызова:***

$curl = curl\_init();

curl\_setopt\_array($curl, **array**(

  CURLOPT\_URL => 'http://exbico/public/calc',

  CURLOPT\_RETURNTRANSFER => **true**,

  CURLOPT\_ENCODING => '',

  CURLOPT\_MAXREDIRS => 10,

  CURLOPT\_TIMEOUT => 0,

  CURLOPT\_FOLLOWLOCATION => **true**,

  CURLOPT\_HTTP\_VERSION => CURL\_HTTP\_VERSION\_1\_1,

  CURLOPT\_CUSTOMREQUEST => 'POST',

  CURLOPT\_POSTFIELDS =>'{

    "intParam" : 2,

    "arParams" : [4, 4, 4, 4, 2, 4, 4]

}',

  CURLOPT\_HTTPHEADER => **array**(

    'Content-Type: application/json'

  ),

));

$response = curl\_exec($curl);

curl\_close($curl);

**echo** $response;

1. **Выход.**

Производится посредством отправки POST запроса на относительный адрес “/logout”.

***Пример вызова:***

$curl = curl\_init();

curl\_setopt\_array($curl, **array**(

  CURLOPT\_URL => 'http://exbico/public/logout',

  CURLOPT\_RETURNTRANSFER => **true**,

  CURLOPT\_ENCODING => '',

  CURLOPT\_MAXREDIRS => 10,

  CURLOPT\_TIMEOUT => 0,

  CURLOPT\_FOLLOWLOCATION => **true**,

  CURLOPT\_HTTP\_VERSION => CURL\_HTTP\_VERSION\_1\_1,

  CURLOPT\_CUSTOMREQUEST => 'POST',

));

$response = curl\_exec($curl);

curl\_close($curl);

**echo** $response;

1. **Запуск скрипта через консоль.**

Запуск через консоль осуществляется без предварительной авторизации посредством выполнения команды “php artisan calc:send” с обязательными параметрами *“intParam”* (число), *“arParams”* (массив), а так же опцией *“--user”* (id пользователя).

Таким образом, при вызове команды по порядку передаются числа через пробел, первое из которых будет интерпретироваться, как значение параметра “intParam”, а все последующие, как значения элементов массива “arParams”, переданные по порядку. При необходимости, id пользователя передаётся после параметров с указанием префикса, например “--user=23” (для пользователя с id 27). При отсутствии идентификатора запись в базу данных будет занесена с указанием id пользователя «0».

***Примеры вызова:***

*С привязкой к пользователю:*

$ php artisan calc:send 5 3 4 3 4 5 6 5 6 --user=1

Здесь переданы параметры:

intParam = 5;

arParams = [3, 4, 3, 4, 5, 6, 5, 6];

--user = 1;

*Без привязки к пользователю:*

$ php artisan calc:send 5 2 4 1 4 8 6 5 8

Здесь переданы параметры:

intParam = 5;

arParams = [2, 4, 1, 4, 8, 6, 5, 8];

***Техническая часть***

Используется фреймворк Laravel.

Пути:

**Контроллеры:** */app/Http/Controllers/*

**Модели:** */app/*

**Файлы миграций:** /*database/migrations/*

**Сидеры:** */database/seeds/*

**Шаблоны:** */resources/views/*

**Команды:** */app/Console/Commands/*

**Файл класса CSRF верификации:** */app/Http/Middleware/VerifyCsrfToken.php*

Обращение по относительному адресу “*/calc*” обрабатывается методом *store* контроллера *CalcController* c передачей в качестве входных параметров запроса и объекта класса Calc одноимённой модели. Логика контроллера обеспечивает запись в таблицу *“calcs”* базы данных:

параметров запроса - числа **intParam** (в *task\_int*) и массива **arParams** (сериализованным массивом в *task\_array*), а так же ответа в виде индекса элемента массива, следующего за разделителем (в *answer*) и id пользователя (в *user\_id*).

Ответ определяется методом *getTaskAnswer* класса *Calc*, вызванным из контекста объекта модели, переданного в контроллер.

Ответ выводится в шаблоне *calc.blade.php* в виде целого числа при запуске через REST, при запуске через консоль – строкой в окно консоли.

Обращения по относительным адресам “/login” и “/logout” обрабатываются методами “login” и “logout” контроллера UserController соответственно.

Именованные роуты *“calc”, ”login”* и *“logout”* вынесены в исключения в файле *VerifyCsrfToken.php*

Для хранения данных о пользователях предназначена таблица *users*, добавлен тестовый пользователь *Sergey*. Таблицы созданы посредством миграций, пользователь добавлен через соответствующий *Seeder*.