Отчет

Скриншоты логирования и описание средств безопасности в конце отчета.

1. Logout

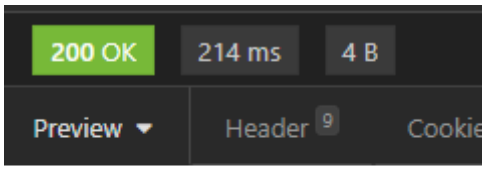
Описание интерфейса:

```php
session_start();
if ($_SERVER['REQUEST_METHOD'] == 'POST'){
    if ($_SESSION['is_auth'] == false){
        die('Already logged out'.PHP_EOL);
    }
    unset($_SESSION);
    session_destroy();
    session_start();
    setcookie("PHPSESSID", session_id(), time()+3600,
        $httponly = true, $domain = 'localhost');
    $_SESSION['is_auth'] = false;
    echo ('OK'.PHP_EOL);
    die(http_response_code(200));
}
else{
    error_log('logout: wrong request method'.PHP_EOL, 3, $destinati
    die(http_response_code(400));
}
```

Начинаем сессию, проверяем метод запроса, сбрасываем массив сессии, сбрасываем сессию, начинаем сессию, создаем новый куки, указываем в массив сессии, что не авторизованы.

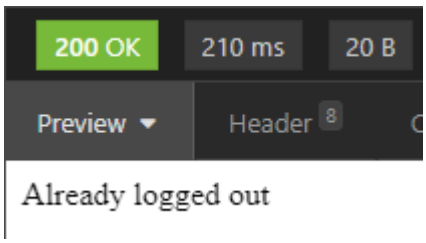
Вызов – logout.php.

Ответ, когда залогинены:



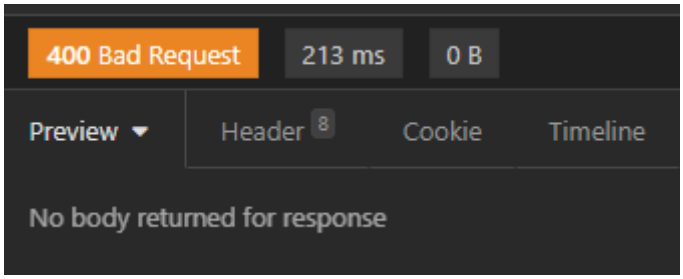
OK

Ответ, когда уже разлогинены:



Обработка ошибок:

1. Неправильный метод запроса



Логирование в лог ошибок

2. **Login**

Описание интерфейса:

```php
<?php

session_start();
setcookie('PHPSESSID', session_id(), time()+3600, '/', $domain = 'localhost', $httponly = true);

try {
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        if (!array_key_exists('login', $_POST)
            || empty($_POST['login'])) {
            $_SESSION['is_auth'] = false;
            error_log('login: wrong login type or login has not been specified'.PHP_EOL, 3, $destination = 'log/error.log');
            echo("Invalid credentials" . PHP_EOL);
            die(http_response_code(200));
        }
        if (!array_key_exists('password', $_POST)
            || empty($_POST['password'])) {
            $_SESSION['is_auth'] = false;
            error_log('login: wrong password type or password has not been specified'.PHP_EOL, 3, $destination = 'log/error.log');
            echo("Invalid credentials" . PHP_EOL);
            die(http_response_code(200));
        }
    } else {
        error_log('login: wrong request type'.PHP_EOL, 3, $destination = 'log/error.log');
        die(http_response_code(400));
    }

    $dbh = new PDO('mysql:host=localhost;dbname=db2', 'egor', '1234');
    $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $login = $_POST['login'];
    $password = $_POST['password'];

    $query =
        'SELECT
            *
        FROM
            `users`
        WHERE
            `login` = ?;';
    $sth = $dbh->prepare($query);
    $sth->execute([$login]);
    $info = $sth->fetchAll(PDO::FETCH_ASSOC);
    if ($info == []) {
        error_log('login: inexistent login has been used'.PHP_EOL, 3, $destination = 'log/security.log');
        $_SESSION['is_auth'] = false;
        echo('Invalid credentials' . PHP_EOL);
        die(http_response_code(200));
```

Начинаем сессию, задаем куки, конструкция try: внутри сам login, ловим ошибку PDO. Проверяем метод запроса, внутри проверяем наличие и правильность переданных в запросе параметров. Получаем из таблицы users логин и пароль, соответствующий переданному логину.

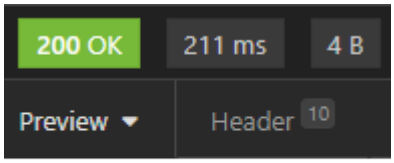
```php
    $serv_password = $info[0]['password'];
    $user_id = $info[0]['user_id'];

    if ($serv_password != null && password_verify($password, $serv_password)){
        if(array_key_exists('is_auth', $_SESSION)){
            if ($_SESSION['is_auth'] == true && $_SESSION['login'] == $login){
                die('You are already logged in'.PHP_EOL);
            }
        }
        session_regenerate_id(true);
        $_SESSION['id'] = session_id();
        $_SESSION['login'] = $login;
        $_SESSION['password'] = $password;
        $_SESSION['user_id'] = $user_id;
        $_SESSION['is_auth'] = true;
        setcookie('PHPSESSID', session_id(), time()+3600, '/',
            $domain = 'localhost', $httponly = true);
        echo('OK'.PHP_EOL);
        die(http_response_code(200));
        }
    else{
        error_log('login: wrong password for user _'.$login.'_ has been used'.PHP_EOL, 3, $destination = 'log/security.log');
        $_SESSION['is_auth'] = false;
        echo('Invalid credentials'.PHP_EOL);
        die(http_response_code(200));
    }
}
catch (PDOException $exception) {
    $_SESSION['is_auth'] = false;
    error_log('login: '.$exception->getMessage().PHP_EOL, 3, $destination = 'log/error.log');
    die(json_encode([]));
}
$dbh = null;
$sth = null;
```

Проверяем правильность пароля с сервера, если верен, то проверяем авторизованы ли мы уже, если нет, то авторизуемся.

Вызов:

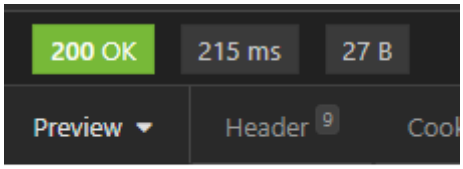
login.php

Ответ, когда еще не залогинены:



OK

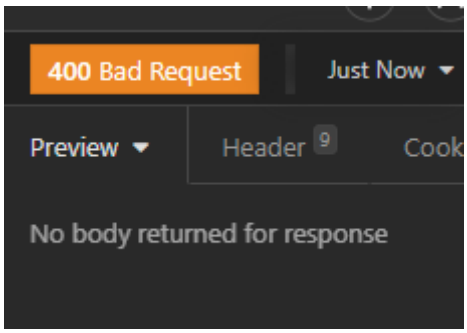
Ответ, когда уже залогинены:



You are already logged in

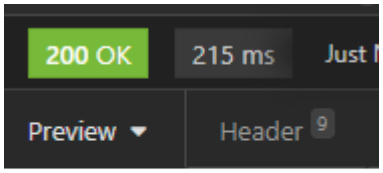
Обработка ошибок.

Неправильный метод запроса. Ответ:



Логирование в лог ошибок

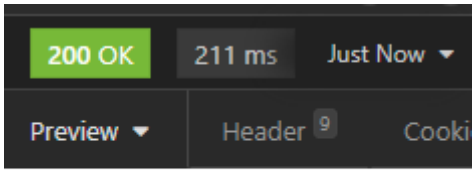
Неправильные параметры. Ответ



Invalid credentials

Логирование в лог ошибок

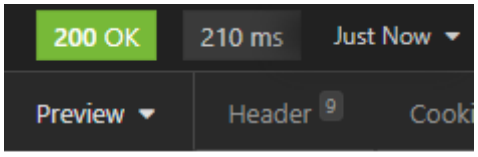
Неправильный логин Ответ



Invalid credentials

Логирование в лог безопасности

Неправильный пароль. Ответ:



Invalid credentials

Логирование в лог безопасности

3. **Public API**

Описание интерфейса:

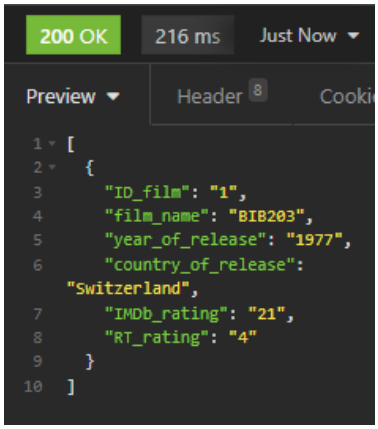
```php
try{
    session_start();
    if ($_SERVER['REQUEST_METHOD'] != 'GET'){
        error_log('public_api: wrong request method'.PHP_EOL, 3, $destination = 'log/error.log');
        die(http_response_code(400));
    }
    if (!array_key_exists('id', $_GET) || empty($_GET['id'])) {
        error_log('public_api: no id specified'.PHP_EOL, 3, $destination = 'log/error.log');
        die(http_response_code(200));
    }

    $dbh = new PDO('mysql:host=localhost;dbname=db2', 'egor', '1234');
    $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $id = $_GET["id"];

    $query =
        'SELECT
            *
        FROM
            `films`
        WHERE
            `ID_film` = ?;';
    $sth = $dbh->prepare($query);
    $sth->execute([$id]);
    $result = $sth->fetchAll(PDO::FETCH_ASSOC);
    echo(json_encode($result));
    die(http_response_code(200));
}
catch (PDOException $exception) {
    $_SESSION['is_auth'] = false;
    error_log('login: '.$exception->getMessage().PHP_EOL, 3, $destination = 'log/error.log');
    die(json_encode([]));
}
$dbh = null;
$sth = null;
```

Стандартный try-catch для ошибок PDO, начинаем сессию, проверяем правильность переданного параметра. Выполняем запрос на выбор записи из таблицы.

Вызов:

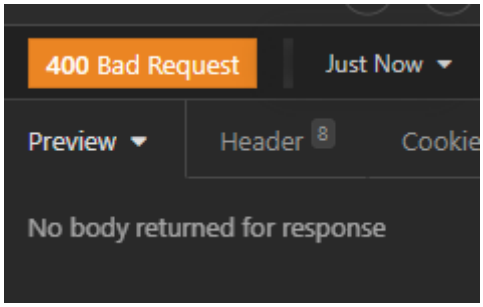
Public_api.php

Ответ:



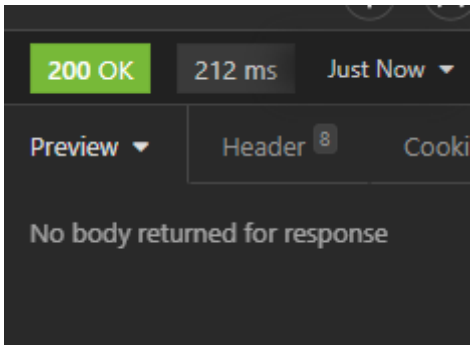
Обработка ошибок:

Неправильный метод:



Логирование в лог ошибок

Неправильные параметры:



Логирование в лог ошибок

4. **Private api**

Описание интерфейса:

```php
<?php
session_start();
try{
    if ($_SERVER['REQUEST_METHOD'] != 'POST'){
        error_log('api: wrong request method', 3, $destination = 'log/error.log');
        die(http_response_code(400));
    }
    if (!array_key_exists('name', $_POST) || empty($_POST['name']) ||
        !array_key_exists('yor', $_POST) || empty($_POST['yor']) ||
        !array_key_exists('cor', $_POST) || empty($_POST['cor']) ||
        !array_key_exists('IMDb', $_POST) || empty($_POST['IMDb']) ||
        !array_key_exists('RT', $_POST) || empty($_POST['RT'])) {
            error_log('api: bad request params'.PHP_EOL, 3, $destination = 'log/error.log');
            die(http_response_code(200));
    }

    $name = $_POST['name'];
    $yor = $_POST['yor'];
    $cor = $_POST['cor'];
    $IMDb = $_POST['IMDb'];
    $RT = $_POST['RT'];

    if ($_SESSION['is_auth'] == true &&
        session_id() == $_COOKIE['PHPSESSID']){
        $dbh = new PDO('mysql:host=localhost;dbname=db2', 'egor', '1234');
        $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $query =
            'INSERT INTO `films`(
            `film_name`,
            `year_of_release`,
            `country_of_release`,
            `IMDb_rating`,
            `RT_rating`
        )
        VALUES(:name, :yor, :cor, :IMDb, :RT);';

        $sth = $dbh->prepare($query);
        $sth->bindParam(':name', $name);
        $sth->bindParam(':yor', $yor);
        $sth->bindParam(':cor', $cor);
        $sth->bindParam(':IMDb', $IMDb);
        $sth->bindParam(':RT', $RT);
        $sth->execute();
        $result = $sth->fetchAll(PDO::FETCH_ASSOC);
```

Try-catch для поимки PDO ошибок. Проверяем метод запроса, проверяем параметры запроса. Если авторизованы: выполняем запрос на добавление записи в таблицу.

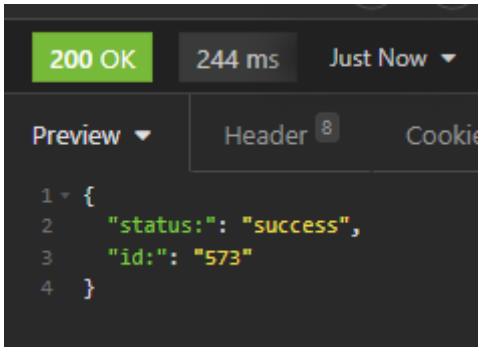
```php
        $query = 'SELECT LAST_INSERT_ID();';
        $sth = $dbh->prepare($query);
        $sth->execute();
        $result = $sth->fetchAll(PDO::FETCH_ASSOC);

        echo json_encode(['status:' =>'success',
            'id:' => $result[0]["LAST_INSERT_ID()"]]);
        die(http_response_code(200));
    }
    else{
        error_log('api: unauthorized attempt to access the table', 3, $destination = 'log/security.log');
        die(http_response_code(403));
    }
}
catch (PDOException $exception) {
    $_SESSION['is_auth'] = false;
    error_log('login: '.$exception->getMessage().PHP_EOL, 3, $destination = 'log/error.log');
    die(json_encode([]));
}
$dbh = null;
$sth = null;
```

Если нет – ошибка.

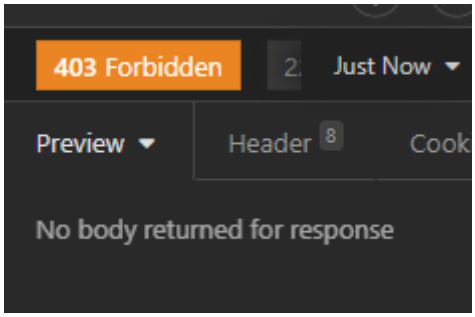
Вызов:

api.php

Ответ, если авторизованы:



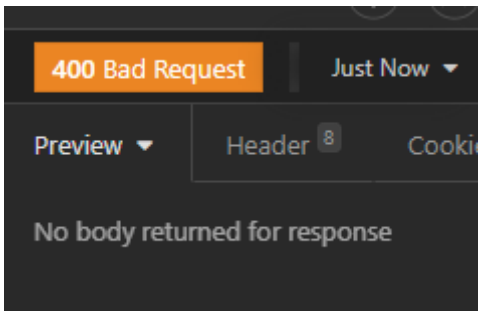
Ответ, если не авторизованы:



Логирование в лог безопасности

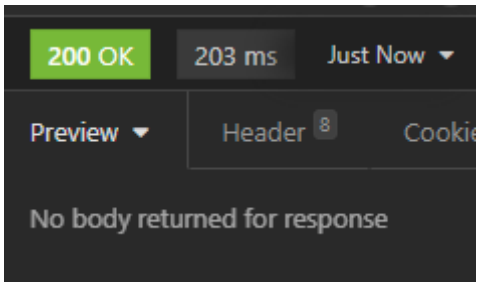
Обработка ошибок:

Неправильный метод запроса:



Логирование в лог ошибок

Неправильные параметры запроса:



Логирование в лог ошибок

5. **Логи**

Лог ошибок:

```
logout: wrong request method
login: wrong request type
login: wrong login type or login has not been specified
public_api: wrong request method
public_api: no id specified
api: wrong request method
api: bad request params
```

Лог безопасности:

```
login: wrong password for user _egor_ has been used
login: inexistent login has been used
api: unauthorized attempt to access the table
```

6. **Безопасность**

Все запросы к БД были экранированы при помощи placeholders (в некоторых случаях безымянных, в некоторых – именных), пароли в базе данных хранятся в виде хеша от пароля, хеш создавался при помощи встроенной функции crypt()