

Отчёт по заданию 5.2 Session management

login.php

Скрипт выглядит следующим образом:

попытка подключиться к базе данных, если безуспешная, то бросается исключение и ошибка записывается в лог:

```
1 <?php
2 try {
3     // создание подключения через connection_string с указанием типа базы
4     $dbh = new PDO( dsn: 'mysql:host=localhost;dbname=db2;charset=utf8', username: 'root', password: 'root');
5     $dbh->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
6 }
7 catch (PDOException $e) {
8     print_r([]);
9     echo PHP_EOL.'Ошибка подключения к БД';
10    $er = $e->getMessage();
11    $log = date( format: 'Y-m-d H:i:s') . ' '.$er;
12    file_put_contents( filename: 'C:/tmp' . '/Log.txt', data: $log. PHP_EOL, flags: FILE_APPEND);
13 }
```

если неверный метод запроса - ошибка 400, тоже записывается в лог:

```
14
15
16 if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
17     http_response_code( response_code: 400);
18     $log = date( format: 'Y-m-d H:i:s') . ' Error 400. Bad Request!';
19     file_put_contents( filename: 'C:/tmp' . '/Log.txt', data: $log. PHP_EOL, flags: FILE_APPEND);
20     //echo 'Error 400 Bad Request.';
21     die();
22 }
23
```

показывается путь для сохранения и дальнейшего хранения сессий, использую режим строгого идентификатора (ID), начинаю сессию, session_set_cookie_params - безопасная настройка свойств cookie, хранящей SID: SameSite защищает от CSRF атак, httponly = true от XSS, lifetime от Session Fixation, secure=true - передача токенов (SID) только по защищенному каналу связи

```
24 ini_set( option: 'session.save_path', value: 'C:/tmp');
25 ini_set( option: 'session.use_strict_mode', value: 1);
26 session_start();
27 session_set_cookie_params([
28     'lifetime' => $maxlifetime = 3600,
29     'path' => '/',
30     'domain' => $_SERVER['HTTP_HOST'],
31     'secure' => $secure = true,
32     'httponly' => $httponly = true,
33     'samesite' => $samesite = 'strict'
34 ]);
```

следующее гарантирует, что никакой web-браузер или промежуточный прокси-сервер не будет кэшировать страницу, таким образом посетители всегда получают самую последнюю версию контента. Фактически, первый заголовок должен быть самодостаточным, это лучший способ гарантировать, что страница не кэшируется. Заголовки Cache-Control и Pragma добавлены с целью «подстраховаться». Хотя они не работают во всех браузерах или прокси, они отловят некоторые случаи, в которых Expires не работает должным образом (например, если дата на компьютере клиента установлена неправильно).

```
35 header( header: 'Expires: Mon, 26 Jul 1997 05:00:00 GMT');
36 header( header: 'Cache-Control: no-store, no-cache, must-revalidate');
37 header( header: 'Cache-Control: post-check=0, pre-check=0', replace: FALSE);
38 header( header: 'Pragma: no-cache');
39
```

если пользователь зарегистрирован, то ему не нужно будет ещё раз это делать:

```
39
40 if( isset($_SESSION['is_auth']) && $_SESSION['is_auth'] ) {
41     echo "You are already authenticated! No need to do it again".PHP_EOL;
42     return;
43 }
44
```

проверяется на ошибки в запросе к базе данных (а также если логин и пароль совпали с теми, что ввёл пользователь, то распознавание пользователя прошло успешно, иначе ошибка фиксируется в логах безопасности):

```

45 try {
46     $query = "SELECT * FROM `registration` WHERE `login` = ? AND `password` = ?";
47     $sth = $dbh->prepare($query);
48     $login = $_POST["login"];
49     $password = $_POST["password"];
50     $request = [$login, $password];
51     $sth->execute($request);
52     $ldb = '';
53     $pdb = '';
54     foreach ($sth as $row) {
55         $ldb = $row[1];
56         $pdb = $row[2];
57     }
58     if( $login == $ldb && $password == $pdb ) {
59         echo "OK".PHP_EOL;
60         $_SESSION['is_auth'] = true;
61     }
62     else {
63         $log = date( format: 'Y-m-d H:i:s') . ' Ошибка авторизации!';
64         file_put_contents( filename: 'C:/tmp' . '/SecurityLog.txt', data: $log. PHP_EOL, flags: FILE_APPEND);
65         echo "Invalid credentials".PHP_EOL;
66     }
67     $dbh = null;
68 }

```

иначе ошибка в лог:

```

69 catch (PDOException $e){
70     print_r([]);
71     echo PHP_EOL.'Ошибка запроса к базе';
72     $er = $e->getMessage();
73     $log = date( format: 'Y-m-d H:i:s') . ' '.$er;
74     file_put_contents( filename: 'C:/tmp' . '/Log.txt', data: $log. PHP_EOL, flags: FILE_APPEND);
75 }

```

Примеры вызова:

правильный запрос:

POST `__base_path/login.php` Send 200 OK 216 ms 4 B

Multipart 2 Auth Query Header 1 Docs

Field	Value
login	user2
password	2222
New name	New value

Raw Header 10 Cookie 1 Timeline

OK

POST `__base_path/login.php` Send 200 OK 233 ms 55 B

Multipart 2 Auth Query Header 1 Docs

Field	Value
login	user2
password	2222
New name	New value

Raw Header 9 Cookie Timeline

You are already authenticated! No need to do it again

некорректные:

GET

base_path/login.php

Send

400 Bad Request

225 ms

0 B

Multipart2

Auth

Query

Header1

Docs

login

user2

password

2222

New name

New value

Raw

Header5

Cookie

Timeline

No body returned for response

POST

base_path/login.php

Send

500 Internal Server Error

236 ms

55 B

Multipart2

Auth

Query

Header1

Docs

login

user2

password

2222

New name

New value

Raw

Header10

Cookie1

Timeline

Array

(

)

Ошибка подключения к БД

POST

base_path/login.php

Send

200 OK

220 ms

51 B

Multipart2

Auth

Query

Header1

Docs

login

user2

password

2222

New name

New value

Raw

Header9

Cookie

Timeline

Array

(

)

Ошибка запроса к базе

POST

base_path/login.php

Send

200 OK

229 ms

21 B

Multipart2

Auth

Query

Header1

Docs

login

user

password

2222

New name

New value

Raw

Header9

Cookie

Timeline

Invalid credentials

logout.php

Скрипт выглядит следующим образом:

проверка метода запроса, если неверный, то ошибка фиксируется в лог файле:

```
1  <?php
2
3  if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
4      http_response_code( response_code: 400);
5      $log = date( format: 'Y-m-d H:i:s') . ' Error 400. Bad Request';
6      file_put_contents( filename: 'C:/tmp' . '/Log.txt', data: $log. PHP_EOL, flags: FILE_APPEND);
7      die();
8  }
```

иначе прописываем путь до сессий, начинаем сессию, если она у нас есть (пользователь зарегистрирован), то удаляем его сессию:

```
9
10  ini_set( option: 'session.save_path', value: 'C:/tmp');
11  session_start();
12  $_SESSION['is_auth'] = true;
13  session_destroy();
14  if (session_id() == ""){
15      echo "OK";
16  }
```

Примеры вызова:

корректный запрос:

The screenshot shows a web client interface with a POST request to `._base_path/logout.php`. The status bar indicates a **200 OK** response with a time of 235 ms and 2 B of data. The 'Raw' tab is selected, showing the response body as `OK`. The URL preview shows `http://localhost:8002/logout.php`.

запрос с ошибкой:

The screenshot shows a web client interface with a PUT request to `._base_path/logout.php`. The status bar indicates a **400 Bad Request** response with a time of 209 ms and 0 B of data. The 'Raw' tab is selected, showing the response body as `No body returned for response`. The URL preview shows `http://localhost:8002/logout.php`.

api.php

Скрипт выглядит следующим образом:

устанавливаю путь, где будет храниться текущая сессия, режим строгого идентификатора (ID) - защита от фиксации сессии, начало сессии:

```
1  <?php
2
3  ini_set( option: 'session.save_path', value: 'C:/tmp');
4  ini_set( option: 'session.use_strict_mode', value: 1);
5
6  session_start();
7
```

неверный метод запроса - исключение:

```
8  if ($_SERVER['REQUEST_METHOD'] !== 'GET') {
9      http_response_code( response_code: 400);
10     $log = date( format: 'Y-m-d H:i:s') . ' Error 400. Bad Request';
11     file_put_contents( filename: 'C:/tmp' . '/Log.txt', data: $log. PHP_EOL, flags: FILE_APPEND);
12     die();
13 }
```

если пользователь авторизован:

проверяется подключение к базе данных, в случае ошибки бросается исключение и записывается в наш файл:

```
14
15  if (isset($_SESSION['is_auth']) && $_SESSION['is_auth']) {
16      echo "OK";
17      try {
18          $dbh = new PDO( dsn: 'mysql:host=localhost;dbname=db2;charset=utf8', username: 'root', password: 'root');
19          $dbh->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
20      }
21      catch (PDOException $e) {
22          print_r([]);
23          echo PHP_EOL.'Ошибка подключения к БД';
24          $er = $e->getMessage();
25          $log = date( format: 'Y-m-d H:i:s') . ' '.$er;
26          file_put_contents( filename: 'C:/tmp' . '/Log.txt', data: $log. PHP_EOL, flags: FILE_APPEND);
27      }
28  }
```

совершение запроса к базе данных, если неправильный запрос - ошибка + она же идёт в лог:

```

28     try {
29         $query = "SELECT * FROM `reader` WHERE Name = ? AND Surname = ?";
30         $sth = $dbh->prepare($query);
31         $name = $_GET["Name"];
32         $surname = $_GET["Surname"];
33         $request = [$name, $surname];
34         $sth->execute($request);
35         $arr = [];
36         foreach ($sth as $row) {
37             $arr['User_id'] = $row[0];
38             $arr['Name'] = $row[1];
39             $arr['Surname'] = $row[2];
40             $arr['Age'] = $row[3];
41             $arr['Location_id'] = $row[4];
42         }
43         $dbh = null;
44     }

45     catch (PDOException $e){
46         print_r([]);
47         echo PHP_EOL.'Ошибка запроса к базе';
48         $er = $e->getMessage();
49         $log = date( format: 'Y-m-d H:i:s') . ' ' . $er;
50         file_put_contents( filename: 'C:/tmp' . '/Log.txt', data: $log. PHP_EOL, flags: FILE_APPEND);
51     }
52 }

```

если пользователь не авторизован, то вылезает ошибка 403:

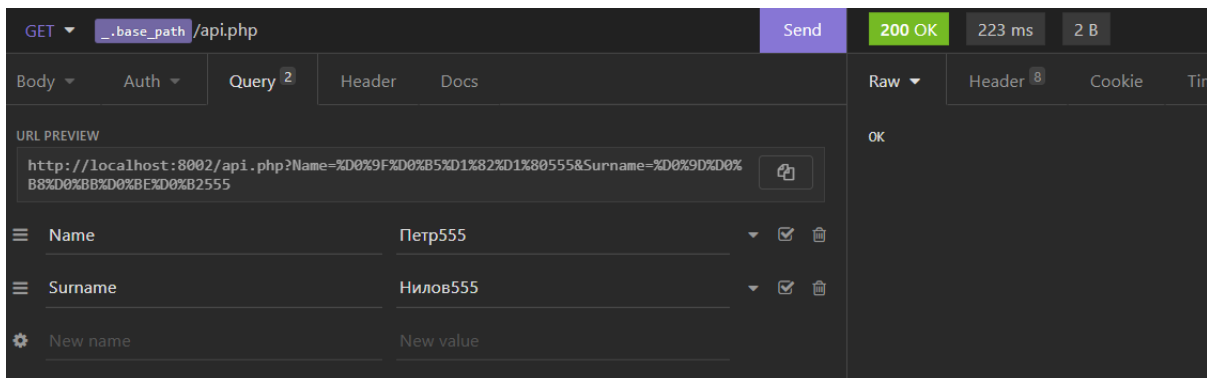
```

53     else{
54         header( header: 'HTTP/1.0 403 Forbidden');
55         echo "Forbidden";
56     }

```

Примеры запуска:

корректный запрос:



The screenshot shows a web browser interface with the following details:

- Method:** GET
- URL:** `http://localhost:8002/api.php`
- Status:** 200 OK
- Response Time:** 223 ms
- Response Size:** 2 B
- Query Parameters:**
 - Name: Петр555
 - Surname: Нилов555
- Response Body:** OK

и некорректные:

GET

Send

200 OK

224 ms

53 B

Body

Auth

Query

Header

Docs

URL PREVIEW

http://localhost:8002/api.php?Name=%D0%9F%D0%B5%D1%82%D1%80555&Surname=%D0%9D%D0%B8%D0%B8%D0%BE%D0%B2555

Name

Петр555

Surname

Нилов555

New name

New value

Raw

Header

Cookie

Timeline

OKArray

(

)

Ошибка запроса к базе

PATCH

Send

400 Bad Request

219 ms

0 B

Body

Auth

Query

Header

Docs

URL PREVIEW

http://localhost:8002/api.php?Name=%D0%9F%D0%B5%D1%82%D1%80555&Surname=%D0%9D%D0%B8%D0%B8%D0%BE%D0%B2555

Name

Петр555

Surname

Нилов555

New name

New value

Raw

Header

Cookie

Timeline

No body returned for response

POST

Send

400 Bad Request

218 ms

0 B

Body

Auth

Query

Header

Docs

URL PREVIEW

http://localhost:8002/api.php?Name=%D0%9F%D0%B5%D1%82%D1%80555&Surname=%D0%9D%D0%B8%D0%B8%D0%BE%D0%B2555

Name

Петр555

Surname

Нилов555

New name

New value

Raw

Header

Cookie

Timeline

No body returned for response

GET

Send

500 Internal Server Error

244 ms

57 B

Body

Auth

Query

Header

Docs

URL PREVIEW

http://localhost:8002/api.php?Name=%D0%9F%D0%B5%D1%82%D1%80555&Surname=%D0%9D%D0%B8%D0%B8%D0%BE%D0%B2555

Name

Петр555

Surname

Нилов555

New name

New value

Raw

Header

Cookie

Timeline

OKArray

(

)

Ошибка подключения к БД

GET

base_path

/api.php

Send

403 Forbidden

218 ms

9 B

Body

Auth

Query²

Header

Docs

Raw

Header⁹

Cookie¹

Timeline

URL PREVIEW

http://localhost:8002/api.php?Name=%D0%9F%D0%B5%D1%82%D1%80555&Surname=%D0%9D%D0%B8%D0%B8%D0%BE%D0%B2555

Name

Петр555

Surname

Нилов555

New name

New value

Forbidden

public-api.php

Скрипт выглядит следующим образом:

идёт подключение к базе данных, если ошибка подключения, то появляется соответствующая ошибка, которая фиксируется в файл Log.txt (находится в C://tmp):

```
1  <?php
2  try {
3      // создание подключения через connection_string с указанием типа базы
4      $dbh = new PDO( dsn: 'mysql:host=localhost;dbname=db2;charset=utf8', username: 'root', password: 'root');
5      $dbh->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
6  }
7  catch (PDOException $e) {
8      print_r([]);
9      echo PHP_EOL.'Ошибка подключения к БД';
10     $er = $e->getMessage();
11     $log = date( format: 'Y-m-d H:i:s') . ' ' . $er;
12     file_put_contents( filename: 'C://tmp' . '/Log.txt', data: $log. PHP_EOL, flags: FILE_APPEND);
13 }
```

идёт проверка метода запроса, если он не соответствует должному, то бросается исключение, а именно метод запроса 400 + фиксируется в файл Log.txt:

```
14 if ($_SERVER['REQUEST_METHOD'] !== 'GET') {
15     http_response_code( response_code: 400);
16     $log = date( format: 'Y-m-d H:i:s') . ' Error 400. Bad Request';
17     file_put_contents( filename: 'C://tmp' . '/Log.txt', data: $log. PHP_EOL, flags: FILE_APPEND);
18     die();
19 }
```

идёт проверка на корректность запроса к базе данных, в случае ошибки это фиксируется в том же файле Log:

```
20 try {
21     $query = "SELECT * FROM 'reader' WHERE Name = ? AND Surname = ?";
22     $sth = $dbh->prepare($query);
23     $name = $_GET["Name"];
24     $surname = $_GET["Surname"];
25     $request = [$name, $surname];
26     $sth->execute($request);
27
28     $arr = [];
29
30     foreach ($sth as $row) {
31         $arr['User_id'] = $row[0];
32         $arr['Name'] = $row[1];
33         $arr['Surname'] = $row[2];
34         $arr['Age'] = $row[3];
35         $arr['Location_id'] = $row[4];
```

```

37     print_r(json_encode($arr));
38 }
39 catch (PDOException $e){
40     print_r([]);
41     echo PHP_EOL.'Ошибка запроса к базе';
42     $er = $e->getMessage();
43     $log = date( format: 'Y-m-d H:i:s') . ' ' . '$er;
44     file_put_contents( filename: 'C:/tmp' . '/Log.txt', data: $log. PHP_EOL, flags: FILE_APPEND);
45 }

```

Примеры вызова:

корректный запрос (любой пользователь может получить информацию о пользователе, для этого он вводит его имя и фамилию)

The screenshot shows a REST client interface with a GET request to `./base_path/public-api.php`. The response is a 200 OK status with a response time of 228 ms and a body size of 129 B. The response body is a JSON object:

```

1 {
2   "User_id": "555",
3   "Name": "Пётр555",
4   "Surname": "Нилов555",
5   "Age": "19",
6   "Location_id": "7"
7 }

```

The query parameters are visible in the 'Query' tab, showing a URL with encoded parameters for Name and Surname.

Ошибки, которые вызываются:

The screenshot shows a REST client interface with a GET request to `./base_path/public-api.php`. The response is a 500 Internal Server Error status with a response time of 209 ms and a body size of 55 B. The response body is an empty array:

```

Array
(
)

```

The error message in the response body is "Ошибка подключения к БД" (Database connection error).

The screenshot shows a REST client interface with a DELETE request to `./base_path/public-api.php`. The response is a 400 Bad Request status with a response time of 221 ms and a body size of 0 B. The response body is empty, with the message "No body returned for response" displayed.

GET ▾

_.base_path

/public-api.php

Send

200 OK

230 ms

51 B

Body ▾

Auth ▾

Query 2

Header

Docs

Raw ▾

Header 5

Cookie

URL PREVIEW

http://localhost:8002/public-api.php?Name=%D0%9F%D0%B5%D1%82%D1%80555&Surname=%D0%9D%D0%B8%D0%BB%D0%BE%D0%B2555

≡

Name

Петр555

▼

✓

🗑

≡

Surname

Нилов555

▼

✓

🗑

⚙

New name

New value




Array

(

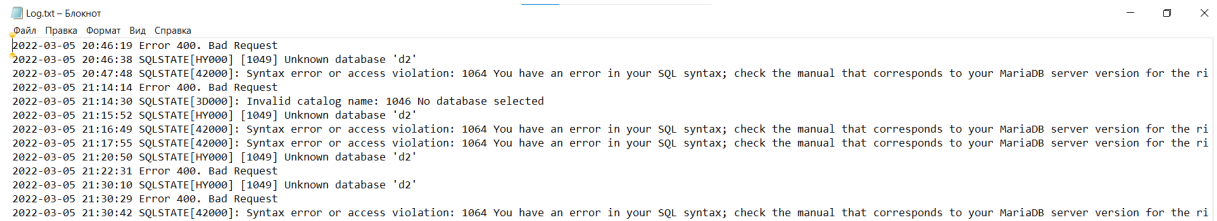
)

Ошибка запроса к базе

Так будут выглядеть данные по указанному пути (мой C:\tmp):

 Log.txt	06.03.2022 0:30	Текстовый докум...	2 КБ
 SecurityLog.txt	06.03.2022 16:18	Текстовый докум...	1 КБ
 sess_a8dtu7uv8smdm4ut0dk4jmogi2	06.03.2022 16:25	Файл	1 КБ

Данные первого файла:



Log.txt – Блокнот

Файл Правка Формат Вид Справка

2022-03-05 20:46:19 Error 400. Bad Request

2022-03-05 20:46:38 SQLSTATE[HY000] [1049] Unknown database 'd2'

2022-03-05 20:47:48 SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the ri

2022-03-05 21:14:14 Error 400. Bad Request

2022-03-05 21:14:30 SQLSTATE[30000]: Invalid catalog name: 1046 No database selected

2022-03-05 21:15:52 SQLSTATE[HY000] [1049] Unknown database 'd2'

2022-03-05 21:16:49 SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the ri

2022-03-05 21:17:55 SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the ri

2022-03-05 21:20:50 SQLSTATE[HY000] [1049] Unknown database 'd2'

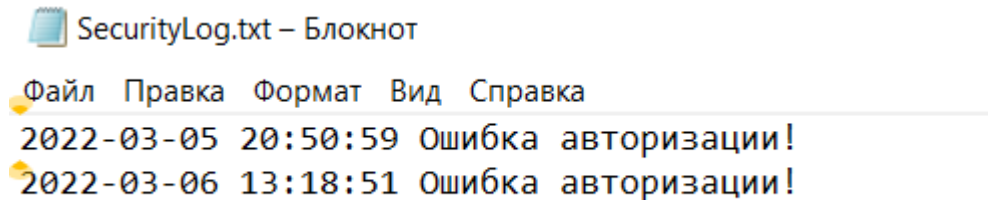
2022-03-05 21:22:31 Error 400. Bad Request

2022-03-05 21:30:10 SQLSTATE[HY000] [1049] Unknown database 'd2'

2022-03-05 21:30:29 Error 400. Bad Request

2022-03-05 21:30:42 SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the ri

Данные второго файла:



SecurityLog.txt – Блокнот











Файл Правка Формат Вид Справка

2022-03-05 20:50:59 Ошибка авторизации!

2022-03-06 13:18:51 Ошибка авторизации!

Все данные зарегистрированных пользователей хранятся в данной таблице в базе данных:

+ Параметры

				reg_id	login	password
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	1	user1	1111
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	2	user2	2222
<input type="checkbox"/>	 Изменить	 Копировать	 Удалить	3	user3	3333