

Отчёт по заданию 5.1 Приложение с базой данных

API listitems.php:

Сначала я создаю подключение к базе данных и задаю генерацию исключения, которое можно перехватить:

```
1 <?php
2
3 try {
4     // создание подключения через connection_string с указанием типа базы
5     $dbh = new PDO( dsn: 'mysql:host=localhost;dbname=bd1;charset=utf8', username: 'root', password: 'root');
6     $dbh->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
7 }
```

Если я не смог подключиться к ней, то генерируется ошибка + её описание:

```
8 catch (PDOException $e) {
9     print_r(json_encode([]));
10    echo PHP_EOL."Ошибка подключения к БД: ". $e->getMessage() . PHP_EOL;
11    //print_r($e); // можно записать подробную информацию об ошибке в лог файл
12    $dbh = null;
13    die();
14 }
```

и программа завершается.

Если метод запроса не GET, то вывожу ошибку и завершаю программу:

```
15 if ($_SERVER['REQUEST_METHOD'] !== 'GET') {
16     echo 'Error 400 Bad Request.';
17     die();
18 }
```

Для генерации ошибок в запросах я создаю строку с генерацией Предупреждений:

```
19 $dbh->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_WARNING);
```

Здесь я собираю информацию о жанре книги, которую передаю в запросе. Если запрос неверный, то вывожу пустой json массив и генерирую ошибку о неправильном запросе к базе данных:

```

20     $query_style = "SELECT * FROM `book` INNER JOIN `style` USING(Style_id) WHERE `Book_name`=?";
21     $sth_style = $dbh->prepare( $query_style );
22     $book_name = $_GET["Book_name"]; // необходимо так же фильтровать значения
23     $sth_style->execute( [$book_name] );
24     if($sth_style !== true) {
25         print_r(json_encode([]));
26         echo PHP_EOL."Ошибка запроса к базе: ".PHP_EOL; print_r($dbh->errorInfo());
27         die();
28     }
29     $arr = [];
30     foreach ($sth_style as $row) {
31         $arr[$book_name.' (style)'] = $row[6];
32     }
33

```

Похожая ситуация с двумя другими характеристиками книги:

```

34     //книга -> год издания
35     $query_recency = "SELECT * FROM `book` INNER JOIN `recency` USING(Recency_id) WHERE `Book_name`=?";
36     $sth_recency = $dbh->prepare( $query_recency );
37     $sth_recency->execute( [$book_name] );
38     if($sth_recency !== true) {
39         print_r(json_encode([]));
40         echo PHP_EOL."Ошибка запроса к базе: ".PHP_EOL; print_r($dbh->errorInfo());
41         die();
42     }
43     foreach ($sth_recency as $row) {
44         $arr[$book_name.' (recency)'] = $row[6];
45     }
46

```

```

47     //книга -> книга на руках у читателя или в библиотеке
48     $query_location = "SELECT * FROM `book` INNER JOIN `location` USING(Location_id) WHERE `Book_name`=?";
49     $sth_location = $dbh->prepare( $query_location );
50     $sth_location->execute([$book_name]);
51     if($sth_location !== true) {
52         print_r(json_encode([]));
53         echo PHP_EOL."Ошибка запроса к базе: ".PHP_EOL; print_r($dbh->errorInfo());
54         die();
55     }
56     foreach ($sth_location as $row) {
57         $arr[$book_name.' (location)'] = $row[6];
58     }
59     print_r(json_encode($arr));
60

```

Следующий закомментированный код выполняет всё то же самое, только для всех книг:

```

62 //выполняем запрос для всех книг -> жанр
63 // $query_all_book_style = "SELECT * FROM `book` INNER JOIN `style` USING(Style_id)";
64 // $res_all_book_style = $dbh->query($query_all_book_style);
65 // if($res_all_book_style !== true) {
66 //     print_r(json_encode([]));
67 //     echo PHP_EOL."Ошибка запроса к базе: ".PHP_EOL; print_r($dbh->errorInfo());
68 //     die();
69 // }
70 // $arrAll = [];
71 // foreach ($res_all_book_style as $row) {
72 //     $arrAll[$row[2].' (style)'] = $row[5];
73 // }
74 // //print_r(json_encode($arrAll));

```

```

76 // //выполняем запрос для всех книг -> год издания
77 // $query_all_book_recency = "SELECT * FROM `book` INNER JOIN `recency` USING(Recency_id)";
78 // $res_all_book_recency = $dbh->query($query_all_book_recency);
79 // if($res_all_book_recency !== true) {
80 //     print_r(json_encode([]));
81 //     echo PHP_EOL."Ошибка запроса к базе: ".PHP_EOL; print_r($dbh->errorInfo());
82 //     die();
83 // }
84 // foreach ($res_all_book_recency as $row) {
85 //     $arrAll[$row[2].' (recency)'] = $row[5];
86 // }
87 // //print_r(json_encode($arrAll));
88 //

```

```

89 // //выполняем запрос для всех книг -> книга на руках у читателя или в библиотеке
90 // $query_all_book_location = "SELECT * FROM `book` INNER JOIN `location` USING (Location_id)";
91 // $res_all_book_location = $dbh->query($query_all_book_location);
92 // if($res_all_book_location !== true) {
93 //     print_r(json_encode([]));
94 //     echo PHP_EOL."Ошибка запроса к базе: ".PHP_EOL; print_r($dbh->errorInfo());
95 //     die();
96 // }
97 // foreach ($res_all_book_location as $row) {
98 //     $arrAll[$row[2].' (location)'] = $row[5];
99 // }
100 // print_r(json_encode($arrAll));
101

```

И освобождаем ресурс:

```

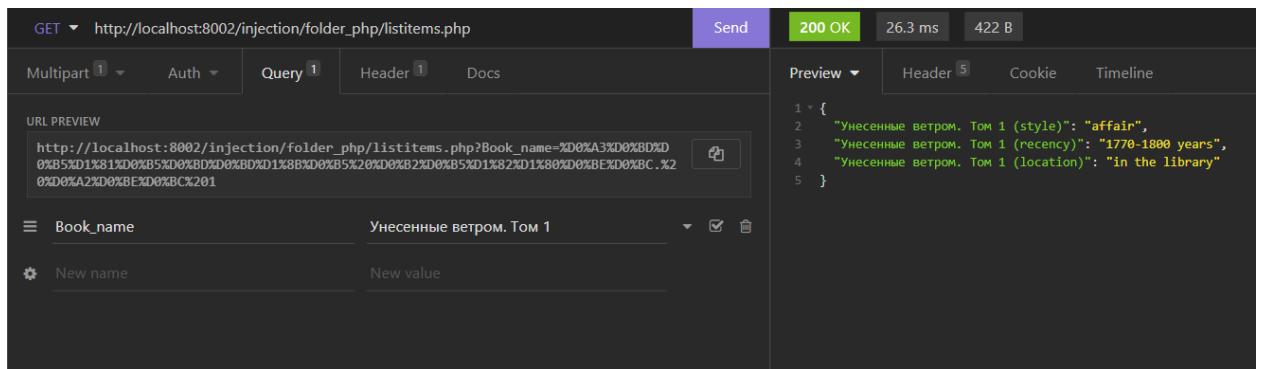
103 $dbh = null;

```

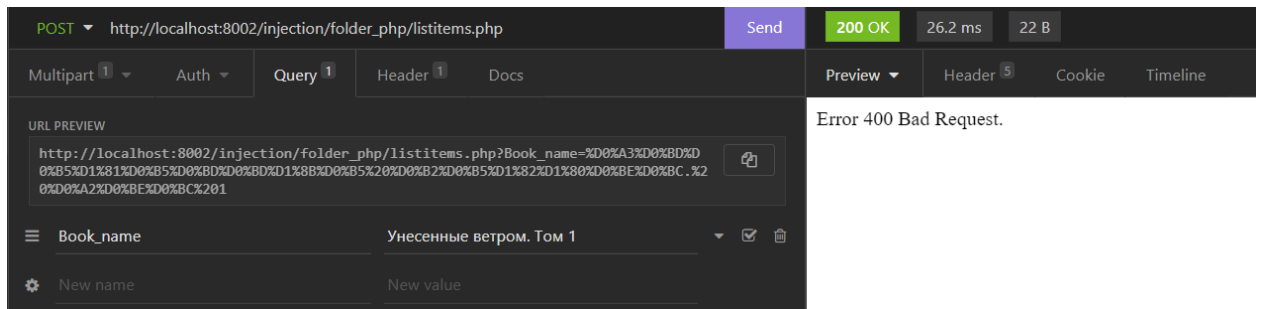
Пример вызова listitems.php и последующие API буду производить с помощью Insomnia.

При правильном вызове

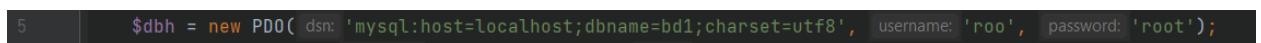
([http://localhost:8002/injection/folder_php/listitems.php?Book_name=%D0%A3%D0%BD%D0%B5%D1%81%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5%20%D0%B2%D0%B5%D1%82%D1%80%D0%BE%D0%BC.%20%D0%A2%D0%BE%D0%BC%201\)](http://localhost:8002/injection/folder_php/listitems.php?Book_name=%D0%A3%D0%BD%D0%B5%D1%81%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5%20%D0%B2%D0%B5%D1%82%D1%80%D0%BE%D0%BC.%20%D0%A2%D0%BE%D0%BC%201):)):



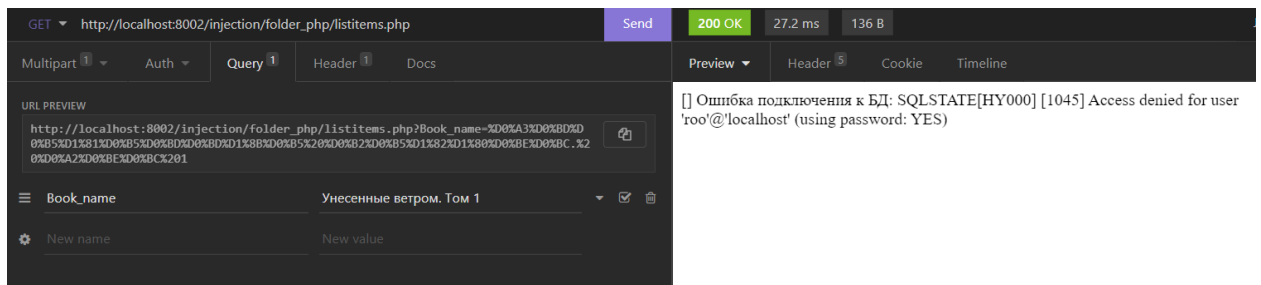
Если другой метод запроса:



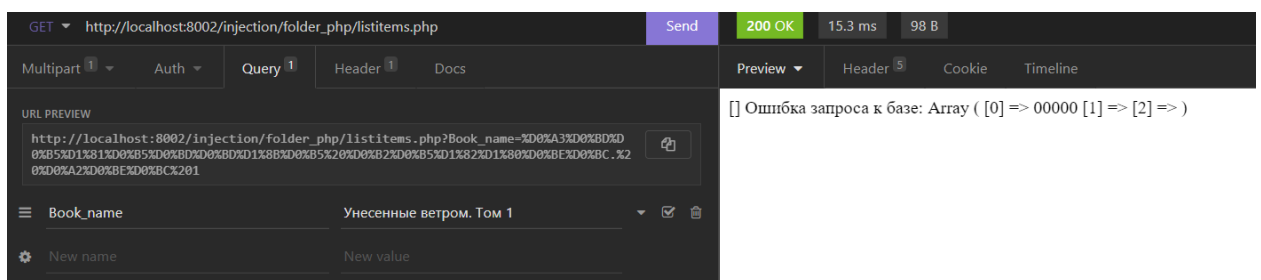
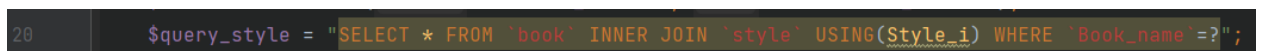
Неправильный пользователь при подключении к базе данных:



Ошибка:

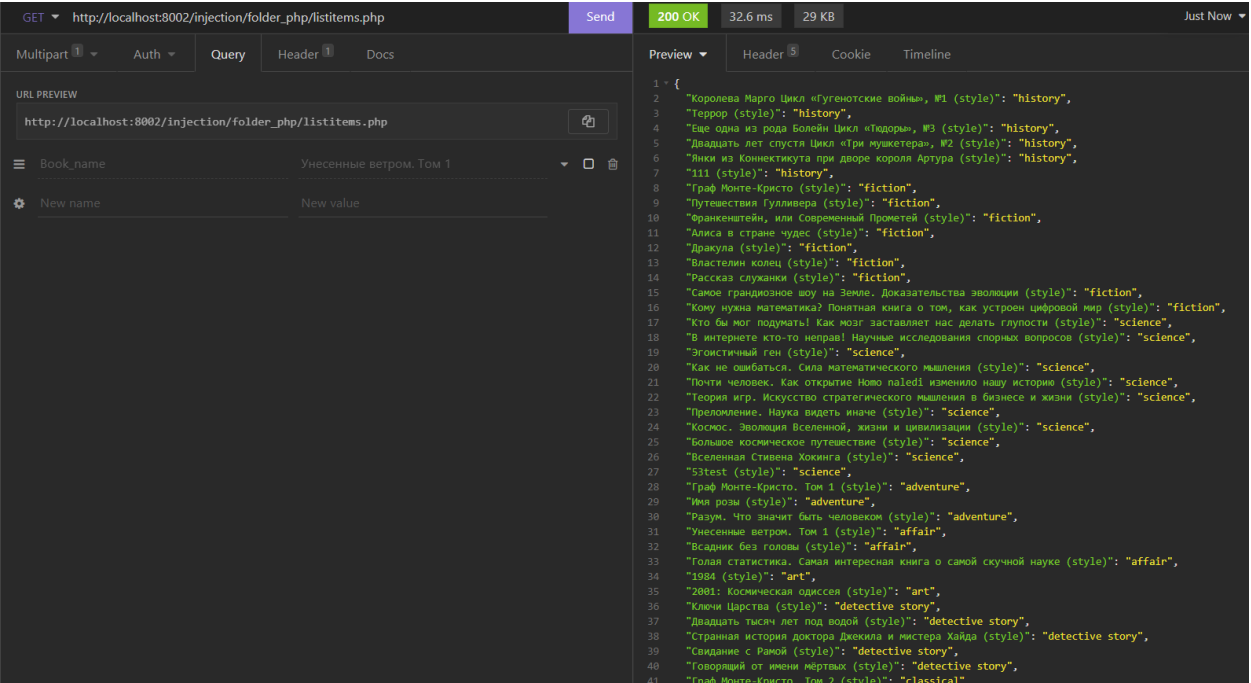


Неправильный запрос (Style_id быть):



Благодаря созданию массива на 29 строке и дальнейшего добавления информации в него, при других ошибках он будет выводиться пустым (в формате json)

Все записи (закомментированный код):

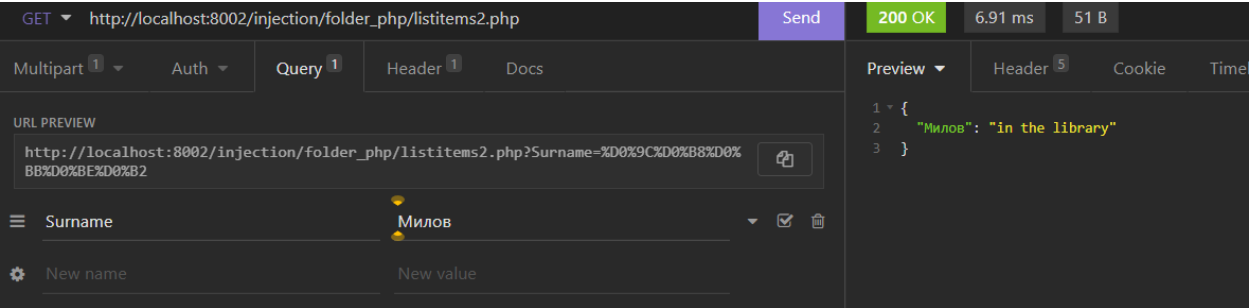


```
1 {
2   "Королева Марго Цикл «Гугенотские войны», #1 (style)": "history",
3   "Terror (style)": "history",
4   "Еще одна из рода Боевин Цикл «Тедоры», #3 (style)": "history",
5   "Двадцать лет спустя Цикл «Гри мушкетера», #2 (style)": "history",
6   "Анна из Коннектикута при дворе короля Артура (style)": "history",
7   "111 (style)": "history",
8   "Граф Монте-Кристо (style)": "fiction",
9   "Путешествия Гулливера (style)": "fiction",
10  "Франкенштейн, или Современный Прометей (style)": "fiction",
11  "Алиса в стране чудес (style)": "fiction",
12  "Дракула (style)": "fiction",
13  "Властелин колец (style)": "fiction",
14  "Рассказ служанки (style)": "fiction",
15  "Самое грандиозное шоу на Земле. Доказательства эволюции (style)": "fiction",
16  "Кому нужна математика? Понятная книга о том, как устроен цифровой мир (style)": "fiction",
17  "Кто бы мог подумать! Как мозг заставляет нас делать глупости (style)": "science",
18  "В интернете кто-то неправа! Научные исследования спорных вопросов (style)": "science",
19  "Эгоистичный ген (style)": "science",
20  "Как не ошибиться: Сила математического мышления (style)": "science",
21  "Почти человек: Как открытие Homo naledi изменило нашу историю (style)": "science",
22  "Теория игр. Искусство стратегического мышления в бизнесе и жизни (style)": "science",
23  "Прелюбление. Наука видеть иначе (style)": "science",
24  "Космос. Эволюция Вселенной, жизни и цивилизации (style)": "science",
25  "Большое космическое путешествие (style)": "science",
26  "Вселенная Стивена Хокинга (style)": "science",
27  "53test (style)": "science",
28  "Граф Монте-Кристо. Том 1 (style)": "adventure",
29  "Имя розы (style)": "adventure",
30  "Разум. Что значит быть человеком (style)": "adventure",
31  "Унесенные ветром. Том 1 (style)": "affair",
32  "Всадник без головы (style)": "affair",
33  "Голая статистика. Самая интересная книга о самой скучной науке (style)": "affair",
34  "1984 (style)": "art",
35  "2001: Космическая одиссея (style)": "art",
36  "Ключи Царства (style)": "detective story",
37  "Двадцать тысяч лет под водой (style)": "detective story",
38  "Странная история доктора Джекила и мистера Хайда (style)": "detective story",
39  "Свидание с Рамой (style)": "detective story",
40  "Говорящий от имени мертвых (style)": "detective story",
41  "Граф Монте-Кристо. Том 2 (style)": "classical"
```

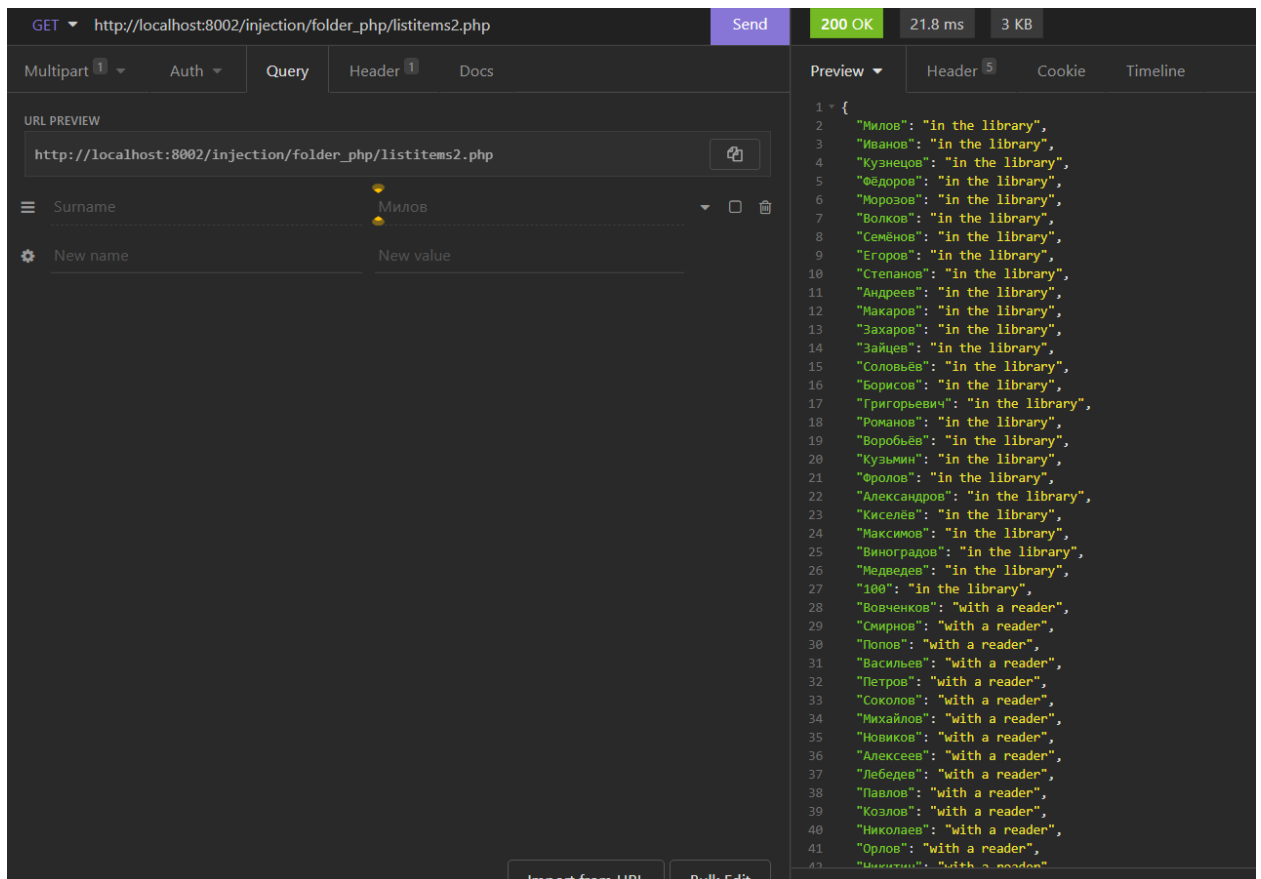
API listitems2.php:

(http://localhost:8002/injection/folder_php/listitems2.php?Surname=%D0%9C%D0%B8%D0%BB%D0%BE%D0%B2)

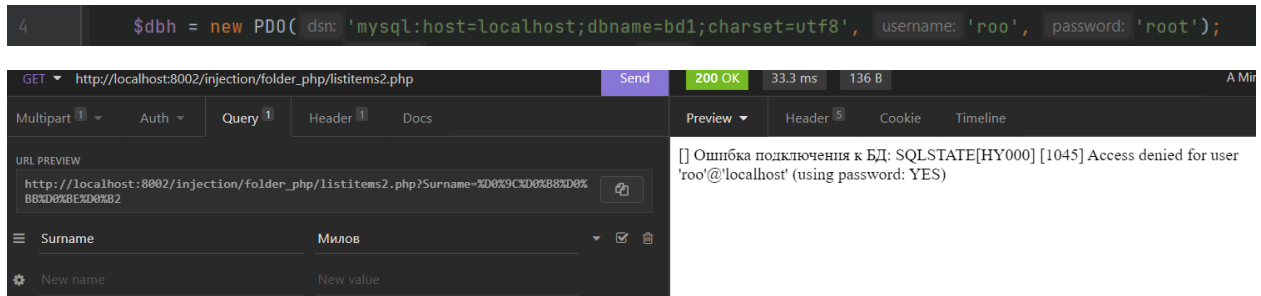
Картина аналогична предыдущему файлу, только вместо названия книги надо вводить фамилию читателя, который зарегистрирован в библиотеке (Запрашивается фамилия и выводится результат, есть ли книга у него на руках или она в библиотеке)



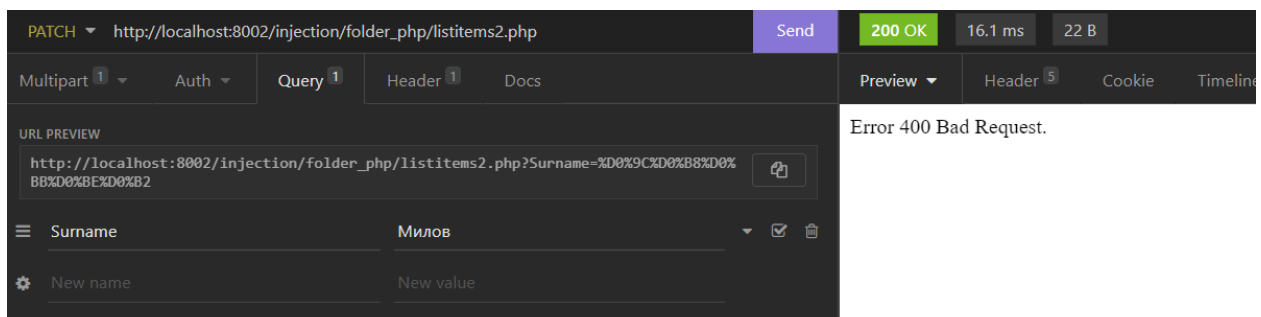
```
1 {
2   "Миров": "in the library"
3 }
```



Ошибка при подключении к базе данных:

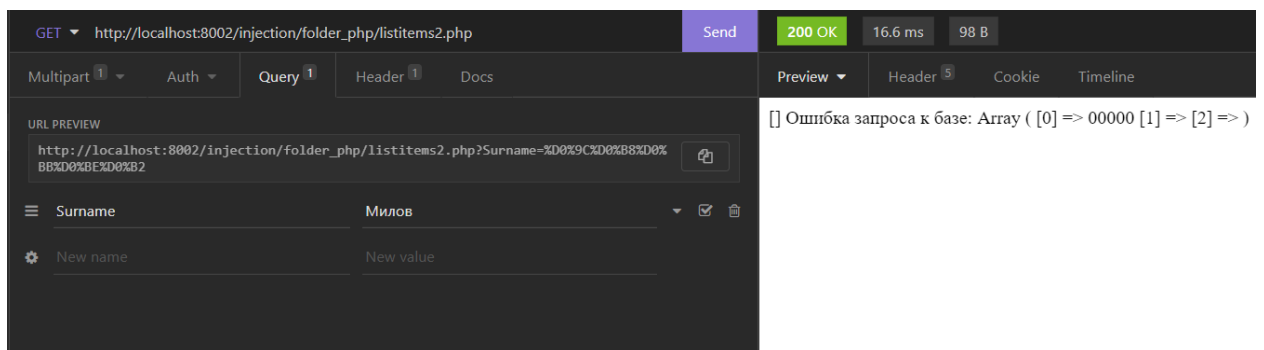


Неверный метод запроса:



Плохой запрос к бд:

```
//читатель -> книга на руках у читателя или в библиотеке
$query_location = "SELECT * FROM `reader` INNER JOIN `location` USING(Location_i) WHERE `Surname`=?";
$sth_location = $dbh->prepare( $query_location );
$surname = $_GET["Surname"]; // необходимо так же фильтровать значения
$sth_location->execute([$surname]);
if($sth_location !== true) {
    print_r(json_encode([]));
    echo PHP_EOL."Ошибка запроса к базе: ".PHP_EOL; print_r($dbh->errorInfo());
    die();
}
$arr_location = [];
foreach ($sth_location as $row) {
    $arr_location[$surname] = $row[5];
}
print_r(json_encode($arr_location));
```



Так же при других ошибках \$sth_location будет оставаться пустым.

API getitem.php:

(http://localhost:8002/injection/folder_php/getitem.php?Book_id=1)

Все те же самые ошибки:

```
<?php
try {
    // создание подключения через connection_string с указанием типа базы
    $dbh = new PDO( dsn: 'mysql:host=localhost;dbname=bd1;charset=utf8', username: 'root', password: 'root');
    $dbh->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    print_r(json_encode([]));
    echo PHP_EOL."Ошибка подключения к БД: ". $e->getMessage() . PHP_EOL;
    //print_r($e); // можно записать подробную информацию об ошибке в лог файл
    $dbh = null;
    die();
}
if ($_SERVER['REQUEST_METHOD'] !== 'GET') {
    echo 'Error 400 Bad Request.';
    die();
}
```

Генерирую запрос для извлечения свойств книги, название которой мы передаём в запросе:

```

19 //id книги -> её свойства
20 $query_book = "SELECT b.`Book_id`, b.`Book_name`, rec.`time`, s.`Style`, l.`Location`, l.`Location_id`
21 FROM `book` b
22 LEFT JOIN `location` l ON b.`Location_id`=l.`Location_id`
23 LEFT JOIN `style` s ON b.`Style_id`=s.`Style_id`
24 LEFT JOIN `recency` rec ON b.`Recency_id`=rec.`Recency_id` WHERE `Book_id`=?";
25 $sth_book = $dbh->prepare($query_book);
26 $book_id = $_GET["Book_id"]; // необходимо так же фильтровать значения
27 $sth_book->execute( [$book_id] );
28 if($sth_book !== true) {
29     print_r(json_encode([]));
30     echo PHP_EOL."Ошибка запроса к базе: ".PHP_EOL; print_r($dbh->errorInfo());
31     die();
32 }
33 $arr_book = [];
34 $location_var = "";

```

+ массивами отправляем всех читателей, которые могут иметь такой статус, как у этой книги (если книга не находится в библиотеке, то она у кого-то на руках, и возможные люди, у которых есть эта книга, так же передаются в json массиве):

```

33 $arr_book = [];
34 $location_var = "";
35 foreach ($sth_book as $row) {
36     $arr_book["Book_id"] = $row[0];
37     $arr_book["Book_name"] = $row[1];
38     $arr_book["time"] = $row[2];
39     $arr_book["Style"] = $row[3];
40     $arr_book["Location"] = $row[4];
41     $location_var = $row[5];
42 }
43 $query = "SELECT * FROM `reader` WHERE `Location_id`=$location_var";
44 $res = $dbh->query($query);
45 if($res !== true) {
46     print_r(json_encode([]));
47     echo PHP_EOL."Ошибка запроса к базе: ".PHP_EOL; print_r($dbh->errorInfo());
48     die();
49 }

```



```

50     $a = [];
51     $count = 0;
52     foreach ($res as $row) {
53         $count += 1;
54         $a[$count]["id"] = $row[0];
55         $a[$count]["Surname"] = $row[2];
56         $a[$count]["Name"] = $row[1];
57     }
58     $arr_book["linkedRecords"] = $a;
59     print_r(json_encode($arr_book));
60
61     // освобождение ресурса
62     $dbh = null;

```

Правильный запрос:

GET http://localhost:8002/injection/folder_php/getitem.php?Book_id=1 200 OK 18.2 ms 3.4 KB

URL PREVIEW: http://localhost:8002/injection/folder_php/getitem.php?Book_id=1

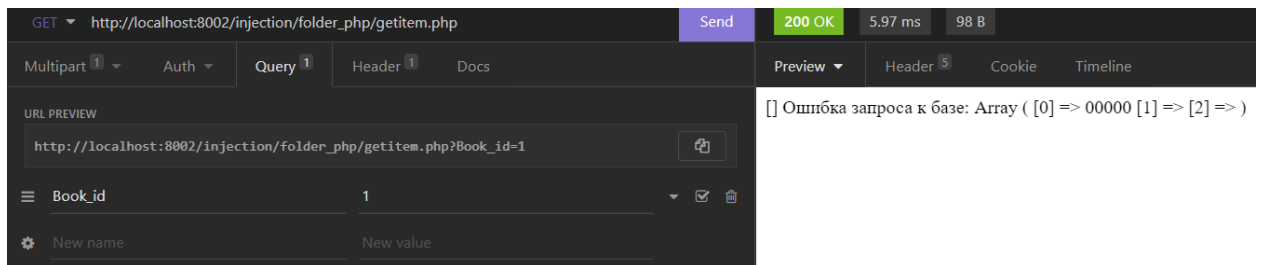
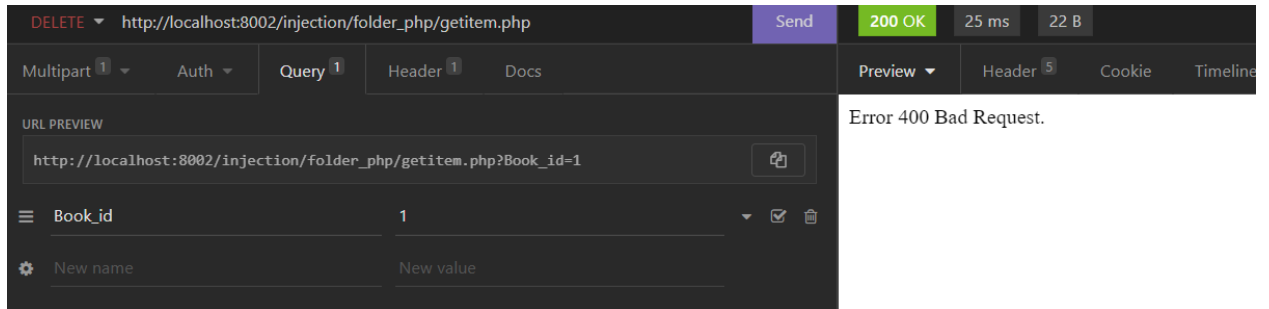
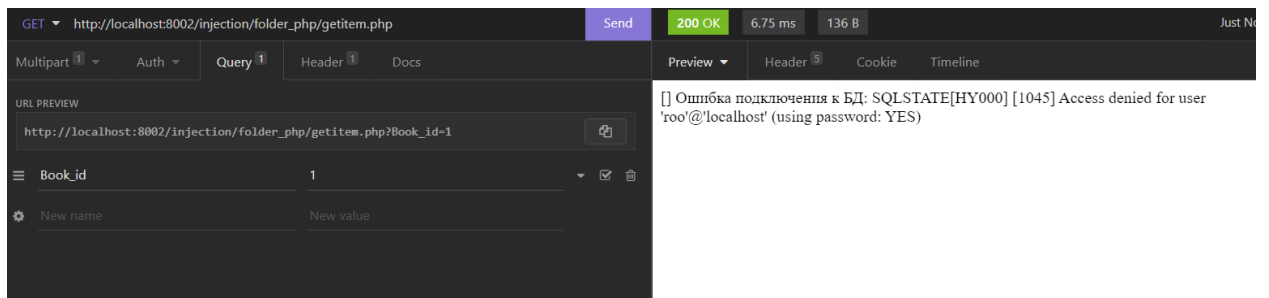
Book_id	1
New name	New value

```

1 {
2   "Book_id": "1",
3   "Book_name": "Унесенные ветром. Том 1",
4   "time": "1770-1800 years",
5   "Style": "affair",
6   "Location": "in the library",
7   "linkedRecords": {
8     "1": {
9       "id": "1",
10      "Surname": "Милов",
11      "Name": "Август"
12    },
13    "2": {
14      "id": "3",
15      "Surname": "Иванов",
16      "Name": "Александр"
17    },
18    "3": {
19      "id": "5",
20      "Surname": "Кузнецов",
21      "Name": "Альберт"
22    },
23    "4": {
24      "id": "12",
25      "Surname": "Фёдоров",
26      "Name": "Никита"
27    },
28    "5": {
29      "id": "13",
30      "Surname": "Морозов",
31      "Name": "Виктор"
32    },
33    "6": {
34      "id": "14",
35      "Surname": "Волков",
36      "Name": "Виталий"
37    },
38    "7": {
39      "id": "17",
40      "Surname": "Семёнов",
41      "Name": "Глеб"
42    }
43  }
44 }

```

И такие же ошибки:



При других ошибках выводится [].

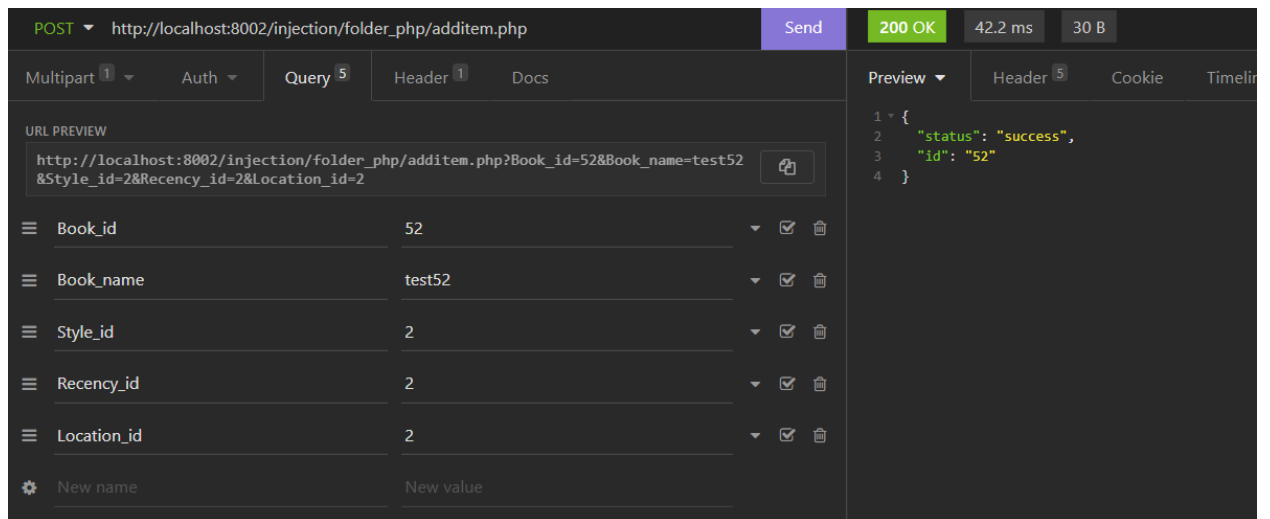
API additem.php:

(http://localhost:8002/injection/folder_php/additem.php?Book_id=52&Book_name=test52&Style_id=2&Recency_id=2&Location_id=2)

Изначально видим нашу последнюю запись:

+ Параметры						
	Book_id	Book_name	Style_id	Recency_id	Location_id	User_id
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	51	Голая статистика. Самая интересная книга о самой с...	5	10	1	NULL
<input type="checkbox"/> Отметить все С отмеченными: <input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить <input type="checkbox"/> Экспорт						

Добавим сразу же после неё новую:

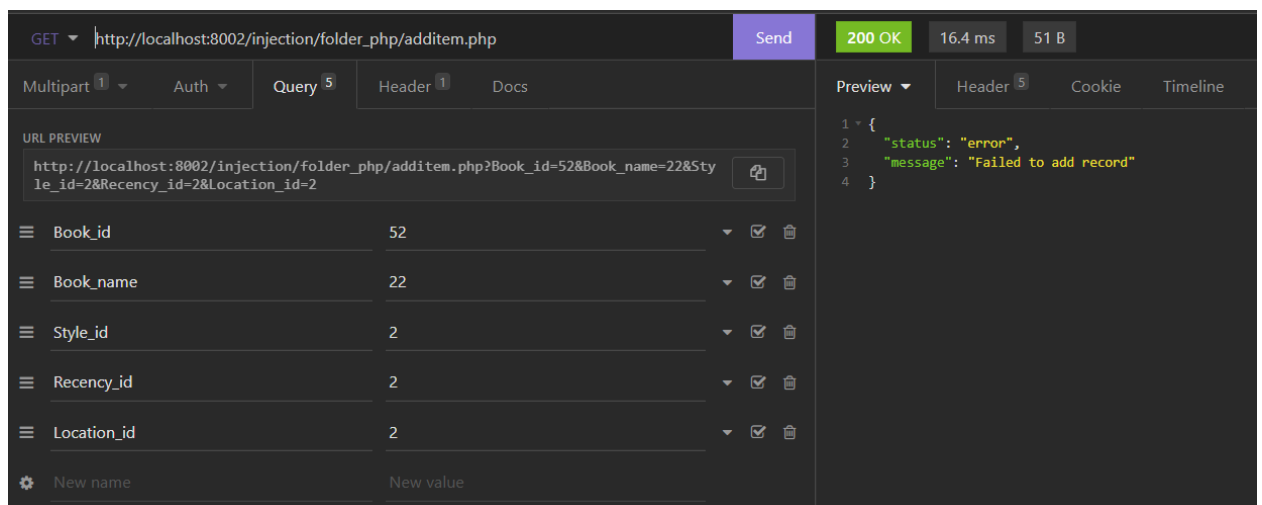


Получили статус «успех» и id новой записи

+ Параметры

	Book_id	Book_name	Style_id	Recency_id	Location_id	User_id
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	51	Голая статистика. Самая интересная книга о самой с...	5	10	1	NULL
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	52	test52	2	2	2	NULL

Если отправить неверным методом, сгенерировать неправильный запрос или сделать какую-либо другую ошибку, будет следующее:



Код программы: сначала мы подключаемся к базе данных, вводим данные для дальнейшего добавления записи и генерируем ошибку на случай неверного метода запроса:

```

1  <?php
2
3  try {
4      $dbh = new PDO( dsn: 'mysql:host=localhost;dbname=bd1;charset=utf8', username: 'root', password: 'root');
5      var_export($dbh->errorinfo());
6      //добавление записи в book
7      $query = "INSERT INTO `book` (`Book_id`, `Book_name`, `Style_id`, `Recency_id`, `Location_id`)
8              VALUES (?, ?, ?, ?, ?)";
9      if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
10         $arr['Error 400 Bad Request'] = "Failed to add record";
11         print_r(json_encode($arr));
12         die();
13     }

```

Далее записываем данные для запроса и в случае добавления записи выводим массив об успешном добавлении:

```

14     // подготовка запроса
15     $sth = $dbh->prepare( $query );
16     // выполнение запроса
17     $Book_id = $_GET["Book_id"];
18     $Book_name = $_GET["Book_name"];
19     $Style_id = $_GET["Style_id"];
20     $Recency_id = $_GET["Recency_id"];
21     $Location_id = $_GET["Location_id"];
22     $book = [$Book_id, $Book_name, $Style_id, $Recency_id, $Location_id]; // необходимо так же фильтровать значения
23     $sth->execute($book);
24     $arr["status"] = "success";
25     $arr["id"] = $Book_id;
26     //если добавилась запись
27     if( $sth === true ) {
28         print_r(json_encode($arr));
29     }
30     // print_r(json_encode($arr));
31     // освобождение ресурса
32     $dbh = null;

```

На случай других ошибок:

```

catch (PDOException $e) {
    $arr['Error 400 Bad Request'] = "Failed to add record";
    print_r(json_encode($arr));
    die();
}

```

API additem2.php

(http://localhost:8002/injection/folder_php/additem2.php?User_id=51&Name=51name&Surname=51sur&Age=51&Location_id=1)

Код такой же, как в предыдущей программе, только меняются переменные в запросе:

```
//добавление записи в reader
$query = "INSERT INTO `reader` (`User_id`, `Name`, `Surname`, `Age`, `Location_id`)
VALUES (?, ?, ?, ?, ?)";

15 $sth = $dbh->prepare( $query );
16 // выполнение запроса
17 $User_id = $_GET["User_id"];
18 $Name = $_GET["Name"];
19 $Surname = $_GET["Surname"];
20 $Age = $_GET["Age"];
21 $Location_id = $_GET["Location_id"];
22 $book = [$User_id, $Name, $Surname, $Age, $Location_id]; // необходимо так же фильтровать значения
23 $sth->execute($book);
24 $arr["status"] = "success";
25 $arr["id"] = $User_id;
26 if( $sth === false ) {
27     echo "Ошибка запроса:".PHP_EOL; print_r($dbh->errorInfo());
28 }
29 else {
30     print_r(json_encode($arr));
31 }
```

Крайние читатели:

<input type="checkbox"/>		Изменить		Копировать		Удалить	49	Роман	Белов	36	2
<input type="checkbox"/>		Изменить		Копировать		Удалить	50	Степан	Медведев	40	1

☐ Отметить все С отмеченными: Изменить Копировать Удалить Экспортировать

<< < 2 > >> ☐ Показать все Количество строк: 25 Фильтровать строки: Поиск в таб

Добавляем запись:

POST
http://localhost:8002/injection/folder_php/additem2.php
Send
200 OK
35.8 ms
30 B

MultipartAuthQueryHeaderDocs

URL PREVIEW

http://localhost:8002/injection/folder_php/additem2.php?User_id=51&Name=51name&Surname=51sur&Age=51&Location_id=1

User_id

51

Name

51name

Surname

51sur

Age

51

Location_id

1

New name

New value

PreviewHeaderCookieTimeline

```

1 {
2   "status": "success",
3   "id": "51"
4 }
```

<< <		3	Показать все		Количество строк: 25	Фильтровать строки:	Поиск
+ Параметры							
			User_id	Name	Surname	Age	Location_id
<input type="checkbox"/>		Изменить		Копировать		Удалить	51 51name 51sur 51 1

Любые ошибки так же выводятся в виде такого json массива:

GET http://localhost:8002/injection/folder_php/additem2.php **Send** 200 OK 23.6 ms 51 B

Multipart 1 Auth Query 5 Header 1 Docs

URL PREVIEW

http://localhost:8002/injection/folder_php/additem2.php?User_id=52&Name=22&Surname=2&Age=2&Location_id=2

User_id	52		
Name	22		
Surname	2		
Age	2		
Location_id	2		

New name New value

Preview

```

1 {
2   "status": "error",
3   "message": "Failed to add record"
4 }
```

POST http://localhost:8002/injection/folder_php/additem2.php **Send** 200 OK 17.9 ms 51 B

Multipart 1 Auth Query 5 Header 1 Docs

URL PREVIEW

http://localhost:8002/injection/folder_php/additem2.php?Another_user=52&Name=22&Surname=2&Age=2&Location_id=2

Another_user	52		
Name	22		
Surname	2		
Age	2		
Location_id	2		

New name New value

Preview

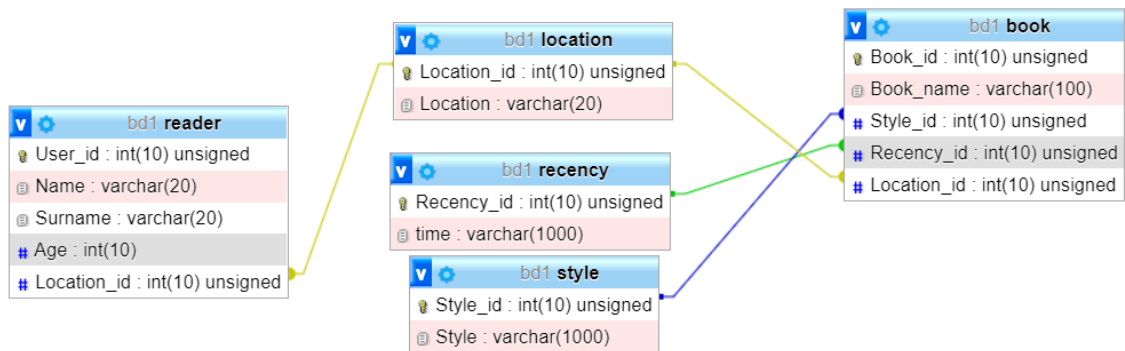
```

1 {
2   "status": "error",
3   "message": "Failed to add record"
4 }
```

API addlink.php

(http://localhost:8002/injection/folder_php/addlink.php?User_id=51&Book_id=52)

Изначально база данных в дизайнерае phpMyAdmin выглядела так:



После запуска текущего API, мы получаем следующее (+ создаём связь между читателем и книгой):

POST `http://localhost:8002/injection/folder_php/addlink.php` **Send** **200 OK** 121 ms 20 B

Multipart **1** Auth **Query** **2** Header **1** Docs

URL PREVIEW

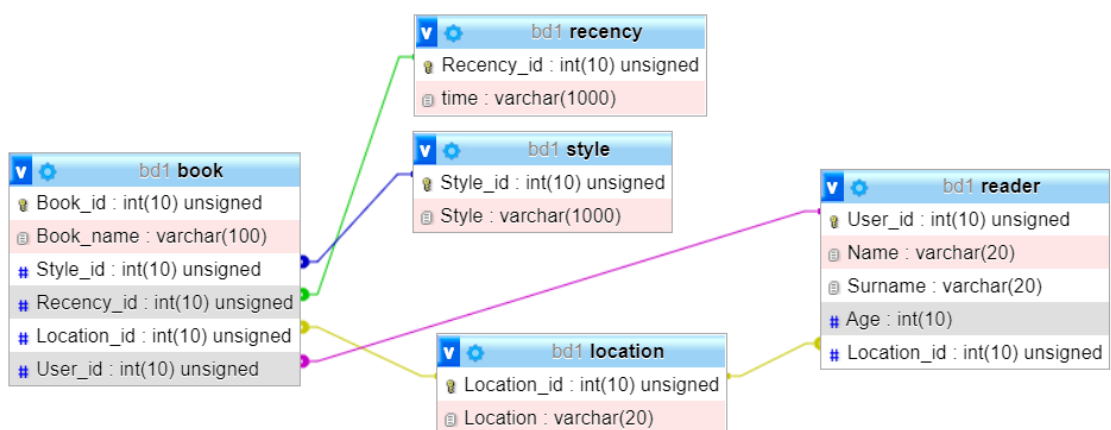
`http://localhost:8002/injection/folder_php/addlink.php?User_id=51&Book_id=52`

Field	Value
User_id	51
Book_id	52
New name	New value

Preview

```

1 {
2   "status": "success"
3 }
  
```



И мы связали две основные сущности: книгу и читателя, которых создали ранее, с помощью их id:

<<

<

3

☐

Показать все

Количество строк: 25

Фильтровать строки:

Поиск в таблице

Сортировать по ключу:

Ниодно

+ Параметры

←

T

→

▼

Book_id

Book_name

Style_id

Recency_id

Location_id

User_id

☐

Изменить

Копировать

Удалить

51

Голая статистика. Самая интересная книга о самой с...

5

10

1

NULL

☐

Изменить

Копировать

Удалить

52

test52

2

2

2

51

↑

☐ Отметить все

С отмеченными:

Изменить

Копировать

Удалить

Экспорт

<<

<

3

☐

Показать все

Количество строк: 25

Фильтровать строки:

+ Параметры

←

T

→

▼

User_id

Name

Surname

Age

Location_id

☐

Изменить

Копировать

Удалить

51

51name

51sur

51

1

Если не удалось подключиться к базе данных, появляется ошибка:

POST http://localhost:8002/injection/folder_php/addlink.php

Send

200 OK 65.3 ms 47 B

Multipart 1 Auth

Query 2

Header 1

Docs

Preview

Header 5

Cookie

Tim

URL PREVIEW

http://localhost:8002/injection/folder_php/addlink.php?User_id=51&Book_id=52

User_id 51

Book_id 52

New name New value

Ошибка подключения к БД

Если метод запроса не POST, вылезает это:

PATCH http://localhost:8002/injection/folder_php/addlink.php

Send

500 Internal Server Error 36.1 ms 22 B

Multipart 1 Auth

Query 2

Header 1

Docs

Preview

Header 5

Cookie

Timeline

URL PREVIEW

http://localhost:8002/injection/folder_php/addlink.php?User_id=51&Book_id=52

User_id 51

Book_id 52

New name New value

Error 400 Bad Request.

Рассмотрим код. Подключение к базе данных + ловлю ошибки и обрабатываю ошибки:


```

1  <?php
2
3  try {
4      // создание подключения через connection_string с указанием типа базы
5      $dbh = new PDO( dsn: 'mysql:host=localhost;dbname=bd1;charset=utf8', username: 'root', password: 'root');
6      $dbh->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
7  }
8  catch (PDOException $e) {
9      print_r(json_encode([]));
10     print PHP_EOL."Ошибка подключения к БД";
11     die();
12 }
13
14 if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
15     echo 'Error 400 Bad Request.';
16 }

```

Создание поля в book, чтобы через него связать две основные сущности, потом создания индекса для ссылки и потом создание самой ссылки между сущностями:

```

10
11 $query_book = "ALTER TABLE `book` ADD `User_id` INT(10) UNSIGNED NULL DEFAULT NULL AFTER `Location_id`";
12 $res_book = $dbh->query($query_book);
13
14
15 $query_index = "ALTER TABLE `book` ADD INDEX(`User_id`)";
16 $res_index = $dbh->query($query_index);
17
18 $query_fk = "ALTER TABLE `book` ADD FOREIGN KEY (`User_id`)
19 REFERENCES `reader`(`User_id`) ON DELETE RESTRICT ON UPDATE RESTRICT";
20 $res_fk = $dbh->query($query_fk);
21

```

Далее получаю два id книги и читателя и связываю, если связалось, то выносятся статус успех:

```

28
29 $query_inf = "UPDATE `book` SET `User_id` = ? WHERE `book`.`Book_id` = ?";
30 $res_inf = $dbh->prepare($query_inf);
31 $data = [$_GET["User_id"], $_GET["Book_id"]]; // необходимо так же фильтровать значения
32 $res_inf->execute($data);
33 $arr["status"] = "success";
34 if( $res_inf === false or $res_book === false or $res_index === false) {
35     print_r(json_encode([]));
36     echo PHP_EOL."Ошибка запроса к базе: ".PHP_EOL; print_r($dbh->errorInfo());
37     die();
38 }
39 print_r(json_encode($arr));
40
41 // освобождение ресурса
42 $dbh = null;

```

API edititem.php

(http://localhost:8002/injection/folder_php/edititem.php?Book_id=100&Book_name=100&Style_id=1&Recency_id=1&Location_id=1&User_id=51&book_id=52)

Изначально в таблице book:

			Book_id	Book_name	Style_id	Recency_id	Location_id	User_id				
<input type="checkbox"/>		Изменить		Копировать		Удалить	51	Голая статистика. Самая интересная книга о самой с...	5	10	1	NULL
<input type="checkbox"/>		Изменить		Копировать		Удалить	52	test52	2	2	2	NULL

Изменение записи:

POST http://localhost:8002/injection/folder_php/edititem.php Send 200 OK 9.73 ms 31 B

Multipart Auth Query Header Docs

URL PREVIEW

http://localhost:8002/injection/folder_php/edititem.php?Book_id=100&Book_name=100&Style_id=1&Recency_id=1&Location_id=1&User_id=51&book_id=52

Book_id 100

Book_name 100

Style_id 1

Recency_id 1

Location_id 1

User_id 51

book_id 52

New name New value

Preview Header Cookie Timeline

1 {

2 "status": "success",

3 "id": "100"

4 }

<< < 3 | ☐ Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице | Сортировать по ключу: Ниод

+ Параметры

Book_id Book_name Style_id Recency_id Location_id User_id

☐

Изменить

Копировать

Удалить

51

Голая статистика. Самая интересная книга о самой с...

5

10

1

NULL

☐

Изменить

Копировать

Удалить

100

100

1

1

1

51

Вызовем одну из ошибок, не задав часть параметров:

POST http://localhost:8002/injection/folder_php/edititem.php Send 200 OK 18.9 ms 52 B

Multipart Auth Query Header Docs

URL PREVIEW

http://localhost:8002/injection/folder_php/edititem.php?Book_id=100&Book_name=100&Style_id=1&Recency_id=1&Location_id=1

Book_id 100

Book_name 100

Style_id 1

Recency_id 1

Location_id 1

New name New value

Preview Header Cookie Timeline

1 {

2 "status": "error",

3 "message": "Failed to edit record"

4 }

Код: создаю подключение к базе данных и генерирую запрос на редактирование поля:

```
1  <?php
2
3  try {
4      // создание подключения через connection_string с указанием типа базы
5      $dbh = new PDO( dsn: 'mysql:host=localhost;dbname=bd1;charset=utf8', username: 'root', password: 'root');
6
7      //книга -> жанр
8      // используются безымянные placeholders - ?
9      $query = "UPDATE `book` SET
10         `Book_id` = ?, `Book_name` = ?, `Style_id` = ?,
11         `Recency_id` = ?, `Location_id` = ?, `User_id` = ?
12         WHERE `book`.`Book_id` = ?;";
13     if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
14         $arr["status"] = "error";
15         $arr["message"] = "Failed to edit record";
16         print_r(json_encode($arr));
17         die();
18     }
```

Если неверный метод запроса, то ошибка:

```
13     if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
14         $arr["status"] = "error";
15         $arr["message"] = "Failed to edit record";
16         print_r(json_encode($arr));
17         die();
18     }
```

Получение новых данных и их вставка в поле + если всё прошло успешно, то вывод успеха:

```
19     // подготовка запроса
20     $sth = $dbh->prepare($query);
21     // выполнение запроса
22     $Book_id = $_GET["Book_id"];
23     $Book_name = $_GET["Book_name"];
24     $Style_id = $_GET["Style_id"];
25     $Recency_id = $_GET["Recency_id"];
26     $Location_id = $_GET["Location_id"];
27     $User_id = $_GET["User_id"];
28     $remain_book_id = $_GET["book_id"];
29     $book = [$Book_id, $Book_name, $Style_id, $Recency_id, $Location_id, $User_id, $remain_book_id];
30     $sth->execute($book);
31
32     $arr["status"] = "success";
33     $arr["id"] = $Book_id;
34     if( $sth === true ) {
35         print_r(json_encode($arr));
36     }
37     //print_r(json_encode($arr));
38
```

Перехват исключений:

```

42     catch (PDOException $e) {
43         $arr["status"] = "error";
44         $arr["message"] = "Failed to edit record";
45         print_r(json_encode($arr));
46         die();
47     }

```

API edititem2.php

(http://localhost:8002/injection/folder_php/edititem2.php?User_id=100&Name=ttt&Surname=ttt&Age=1&Location_id=1&user_id=51)

Изменим ранее созданную запись в reader с id 51 на следующее:

POST http://localhost:8002/injection/folder_php/edititem2.php 200 OK 29.8 ms 31 B

URL PREVIEW
http://localhost:8002/injection/folder_php/edititem2.php?User_id=100&Name=ttt&Surname=ttt&Age=1&Location_id=1&user_id=51

```

1 {
2   "status": "success",
3   "id": "100"
4 }

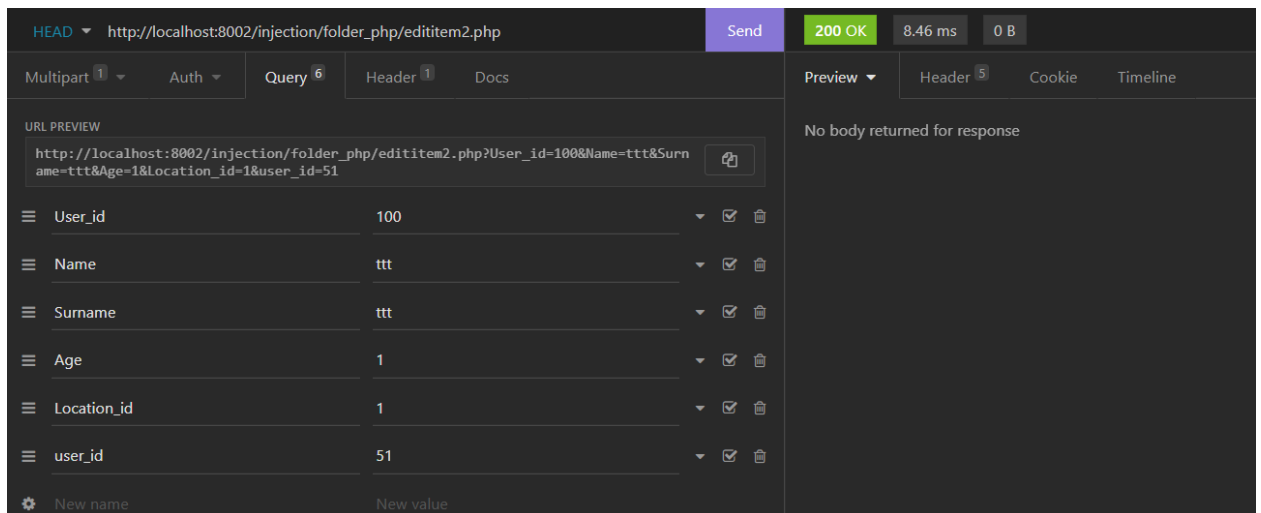
```

Parameter	Value
User_id	100
Name	ttt
Surname	ttt
Age	1
Location_id	1
user_id	51

Параметры

	Book_id	Book_name	Style_id	Recency_id	Location_id	User_id
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	51	Голая статистика. Самая интересная книга о самой с...	5	10	1	NULL
<input type="checkbox"/> Изменить <input type="checkbox"/> Копировать <input type="checkbox"/> Удалить	100	100	1	1	1	NULL

Отправим другим методом (или сделаем какую-то другую ошибку, результат будет один):



Аналогично предыдущему коду, здесь идёт подключение к базе данных, запрос, принятие данных для изменения записи, в случае добавления появляется статус успеха и новый id. В конце кода принимаются исключения:

```

1  <?php
2
3  try {
4      // создание подключения через connection_string с указанием типа базы
5      $dbh = new PDO( dsn: 'mysql:host=localhost;dbname=bd1;charset=utf8', username: 'root', password: 'root');
6
7      //книга -> жанр
8      // используются безымянные placeholders - ?
9      $query = "UPDATE `reader` SET
10         `User_id` = ?, `Name` = ?, `Surname` = ?,
11         `Age` = ?, `Location_id` = ?
12         WHERE `reader`.`User_id` = ?;";
13
14     if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
15         $arr["status"] = "error";
16         $arr["message"] = "Failed to edit record";
17         print_r(json_encode($arr));
18         die();
19     }
20     // подготовка запроса
21     $sth = $dbh->prepare($query);
22     // выполнение запроса
23     $User_id = $_GET["User_id"];
24     $Name = $_GET["Name"];
25     $Surname = $_GET["Surname"];
26     $Age = $_GET["Age"];
27     $Location_id = $_GET["Location_id"];
28     $remain_user_id = $_GET["user_id"];
29     $book = [$User_id, $Name, $Surname, $Age, $Location_id, $remain_user_id]; // необходимо так же фильтровать значения
    $sth->execute($book);

```

```

30
31     $arr["status"] = "success";
32     $arr["id"] = $User_id;
33     if( $sth === true ) {
34         print_r(json_encode($arr));
35     }
36     //print_r(json_encode($arr));
37
38     // освобождение ресурса
39     $dbh = null;
40 }
41 catch (PDOException $e) {
42     $arr["status"] = "error";
43     $arr["message"] = "Failed to edit record";
44     // echo $e->getMessage();
45     print_r(json_encode($arr));
46     die();
47 }

```

API deleteitem.php

(http://localhost:8002/injection/folder_php/deleteitem.php?Book_id=100)

Удалим когда-то созданную запись:

<< < 3

☐ Показать все

Количество строк: 25

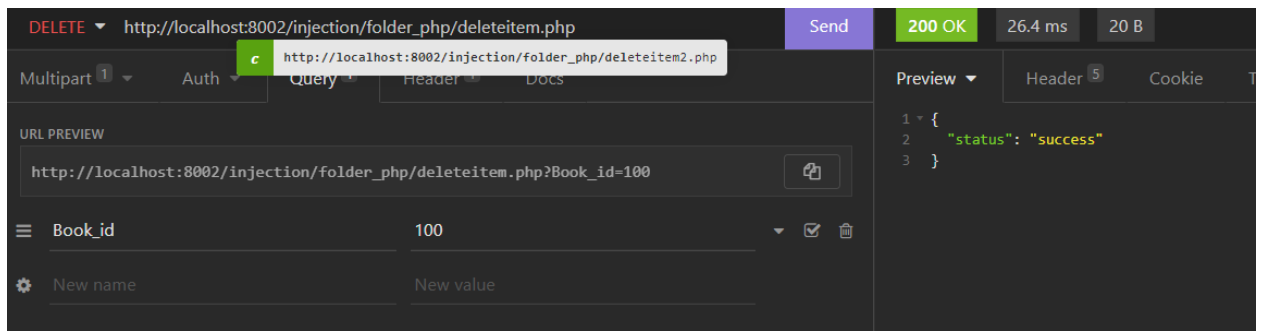
Фильтровать строки:

Сортировать по ключу: Ниодного

+ Параметры

			Book_id	Book_name	Style_id	Recency_id	Location_id	User_id				
<input type="checkbox"/>		Изменить		Копировать		Удалить	51	Голая статистика. Самая интересная книга о самой с...	5	10	1	NULL
<input type="checkbox"/>		Изменить		Копировать		Удалить	100	100	1	1	1	NULL

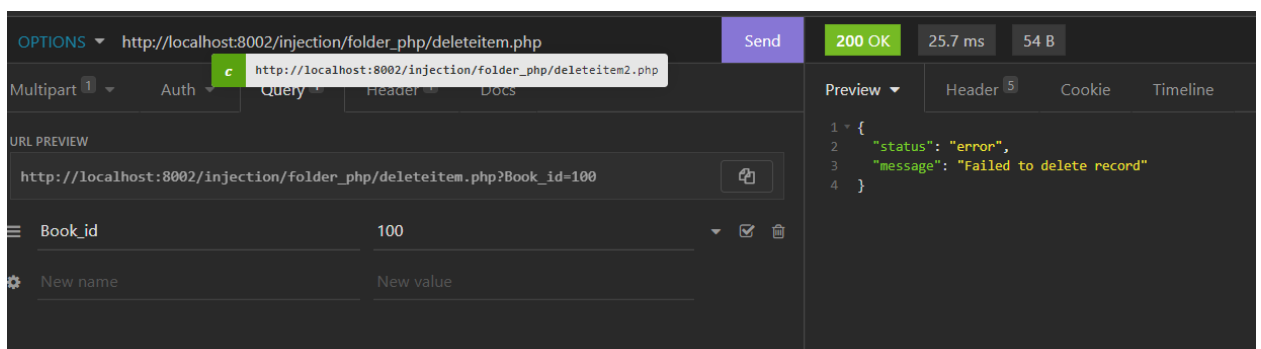
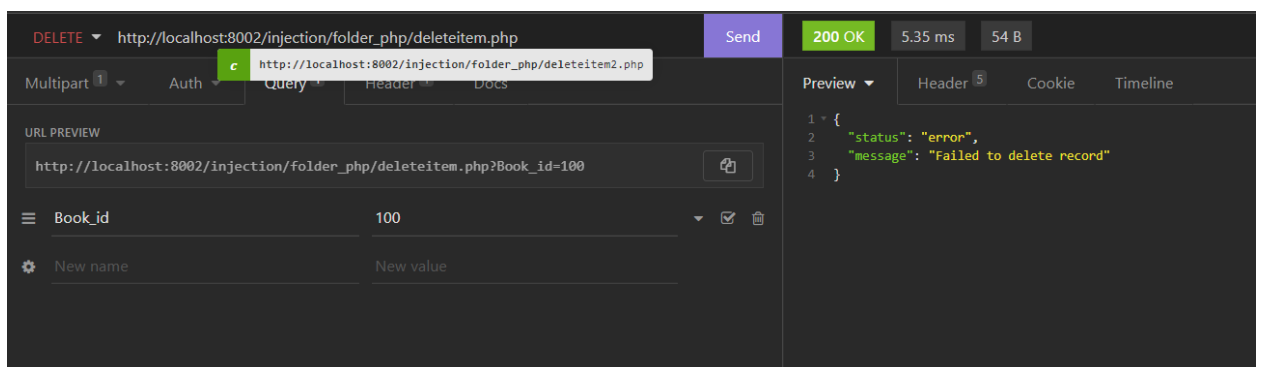
Сгенерируем удаление:



Проверяем запись:

<<	<	3	Показать все	Количество строк: 25	Фильтровать строки:	Поиск в таблице	Сортировать по ключу:	Нисходящего
+ Параметры								
		Book_id	Book_name	Style_id	Recency_id	Location_id	User_id	
<input type="checkbox"/>	Изменить	Копировать	Удалить	51	Голая статистика. Самая интересная книга о самой с...			5 10 1 NULL
Отметить все С отмеченными: Изменить Копировать Удалить Экспорт								

Создадим ошибку при подключении к базе данных или отправим неверным методом запрос:



Как обычно, создаю подключение к базе данных, если ошибки есть, то их обрабатываю, если неверный метод запроса, то же самое. Создаю запрос на удаление записи в таблице. Если удалась, то кидает успех, иначе ошибку удаления. И, конечно, освобождение ресурса.

```

1  <?php
2
3  try {
4      // создание подключения через connection_string с указанием типа базы
5      $dbh = new PDO( dsn: 'mysql:host=localhost;dbname=bd1;charset=utf8', username: 'root', password: 'root');
6      $dbh->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
7  }
8  catch (PDOException $e) {
9      $arr["status"] = "error";
10     $arr["message"] = "Failed to delete record";
11     print_r(json_encode($arr));
12     die();
13 }
14 if ($_SERVER['REQUEST_METHOD'] !== 'DELETE') {
15     $arr["status"] = "error";
16     $arr["message"] = "Failed to delete record";
17     print_r(json_encode($arr));
18     die();
19 }
20 //книга -> жанр
21 $query = "DELETE FROM `book` WHERE `book`.`Book_id` = ?;";
22 $sth = $dbh->prepare($query);
23 $book_id = $_GET["Book_id"]; // необходимо так же фильтровать значения
24 $sth->execute( [$book_id] );

```

```

27     $arr = [];
28     $arr["status"] = "success";
29     //print_r(json_encode($arr));
30
31     if( $sth === true ) {
32         print_r(json_encode($arr));
33     }
34
35     // освобождение ресурса
36     $dbh = null;
37

```

API deleteitem2.php

(http://localhost:8002/injection/folder_php/deleteitem2.php?User id=100)

Изначально мы имеем нашу тестовую запись в таблице reader:

[+ Параметры](#)

	User_id	Name	Surname	Age	Location_id
<input type="checkbox"/> Изменить Копировать Удалить	100	ttt	ttt	1	1

☐ Отметить все
 С отмеченными: Изменить Копировать Удалить

Неверный метод приводит к ошибке:

OPTIONS http://localhost:8002/injection/folder_php/deleteitem2.php Send 200 OK 14.8 ms 54 B

Multipart 1 Auth Query 1 Header 1 Docs

URL PREVIEW

http://localhost:8002/injection/folder_php/deleteitem2.php?User_id=100

User_id 100

New name New value

Preview Header 5 Cookie Timeline

```

1 {
2   "status": "error",
3   "message": "Failed to delete record"
4 }
  
```

Другие ошибки, например, нет соединения с базой данных, так же приводят к этому:

DELETE http://localhost:8002/injection/folder_php/deleteitem2.php Send 200 OK 18.9 ms 54 B

Multipart 1 Auth Query 1 Header 1 Docs

URL PREVIEW

http://localhost:8002/injection/folder_php/deleteitem2.php?User_id=100

User_id 100

New name New value

Preview Header 5 Cookie Timeline

```

1 {
2   "status": "error",
3   "message": "Failed to delete record"
4 }
  
```

Отправляем правильный запрос:

DELETE http://localhost:8002/injection/folder_php/deleteitem2.php Send 200 OK 27 ms 20 B

Multipart 1 Auth Query 1 Header 1 Docs

URL PREVIEW

http://localhost:8002/injection/folder_php/deleteitem2.php?User_id=100

User_id 100

New name New value

Preview Header 5 Cookie Timeline

```

1 {
2   "status": "success"
3 }
  
```

Запись удалена:

<input type="checkbox"/>		Изменить		Копировать		Удалить	48	Роберт	Ковалев	18	2
<input type="checkbox"/>		Изменить		Копировать		Удалить	49	Роман	Белов	36	2
<input type="checkbox"/>		Изменить		Копировать		Удалить	50	Степан	Медведев	40	1

☐ Отметить все
 С отмеченными:
 Изменить
 Копировать
 Удалить
 Э

<< < 2 > >>
 ☐ Показать все
 Количество строк: 25
 Фильтровать строки:
 Поиск в

Код похож на предыдущий: подключаются к базе данных, в случае ошибки подключения выводится ошибка, в неверном методе запроса так же ошибка обрабатывается. Если запись удалена, то выходит успех, иначе ошибка.

```

1  <?php
2
3  try{
4      // создание подключения через connection_string с указанием типа базы
5      $dbh = new PDO( dsn: 'mysql:host=localhost;dbname=bd1;charset=utf8', username: 'root', password: 'root');
6      $dbh->setAttribute( attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
7  }
8  catch (PDOException $e) {
9      $arr["status"] = "error";
10     $arr["message"] = "Failed to delete record";
11     print_r(json_encode($arr));
12     die();
13 }
14
15 if ($_SERVER['REQUEST_METHOD'] !== 'DELETE') {
16     $arr["status"] = "error";
17     $arr["message"] = "Failed to delete record";
18     print_r(json_encode($arr));
19     die();

```

```

21
22     $query = "DELETE FROM `reader` WHERE `reader`.`User_id` = ?";
23     $sth = $dbh->prepare($query);
24     $book_id = $_GET["User_id"]; // необходимо так же фильтровать значения
25     $sth->execute( [$book_id] );
26
27     $arr = [];
28     $arr["status"] = "success";
29     // print_r(json_encode($arr));
30
31     if( $sth === true ) {
32         print_r(json_encode($arr));
33     }
34
35
36     // освобождение ресурса
37     $dbh = null;
38

```

Заполнение базы данных db2 производилась с помощью скриптов fill_books.php, fill_readers.php, fill_recency.php, fill_style.php, fill_location.php.

В основных сущностях (reader, book) по 555 записей, в дополнительных (recency, location, style) по 25.

Для тестирования API следует брать другие данные (Потому что данная база данных является предыдущей работой, ничем не отличающаяся от нужной, кроме числа записей и их названиями), отличные от описанных в примере. Их можно найти в самой базе данных. Например, при поиске данных о читателе, указать фамилию Нилов1 вместо Милов, как было в примере listitems2.php:

1

>

>>

Количество строк: 25

Фильтровать строки:

Поиск в таблице

+ Параметры

←

T

→

▼

	User_id	Name	Surname	Age	Location_id
<div><div><div><div><input type="checkbox"/></div><div></div><div>Изменить</div></div><div><div></div><div>Копировать</div></div><div><div></div><div>Удалить</div></div></div></div>	1	Пётр1	Нилов1	83	4
<div><div><div><div><input type="checkbox"/></div><div></div><div>Изменить</div></div><div><div></div><div>Копировать</div></div><div><div></div><div>Удалить</div></div></div></div>	2	Пётр2	Нилов2	70	5
<div><div><div><div><input type="checkbox"/></div><div></div><div>Изменить</div></div><div><div></div><div>Копировать</div></div><div><div></div><div>Удалить</div></div></div></div>	3	Пётр3	Нилов3	22	9
<div><div><div><div><input type="checkbox"/></div><div></div><div>Изменить</div></div><div><div></div><div>Копировать</div></div><div><div></div><div>Удалить</div></div></div></div>	4	Пётр4	Нилов4	54	4
<div><div><div><div><input type="checkbox"/></div><div></div><div>Изменить</div></div><div><div></div><div>Копировать</div></div><div><div></div><div>Удалить</div></div></div></div>	5	Пётр5	Нилов5	60	4