**Функция авторизации**

function authOPAC() {

require('config.php');

require\_once('grant.php');

$request = [ // формирование запроса на авторизацию

'grant\_type' => $GRANT\_TYPE,

'client\_id' => $CLIENT\_ID,

'client\_secret' => $CLIENT\_SECRET,

'username' => $USERNAME,

'password' => $PASSWORD,

'scope' => $SCOPE,

];

$ch = curl\_init(); // инициализация запроса cURL и формирование его полей

curl\_setopt($ch, CURLOPT\_URL, $URL\_API.$REG\_AUTH); //URL запроса

curl\_setopt($ch, CURLOPT\_POST, 1); // тип запроса (запрос POST)

curl\_setopt($ch, CURLOPT\_POSTFIELDS, http\_build\_query($request)); //формирование текста запроса POST

curl\_setopt($ch, CURLOPT\_HTTPHEADER, array($REG\_CONT\_URL, $RESP\_CONT\_JSON)); // заголовок запроса

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true); //возврат кода результатов передачи

$response = json\_decode(curl\_exec($ch)); //выполнение запроса и декодирование

$httpcode = curl\_getinfo($ch, CURLINFO\_RESPONSE\_CODE); // выделение кода ответа сервера

if ($httpcode == 200) {

print\_r('Код ответа на запрос авторизации = '.$httpcode.'<br>');

/\*

echo "<pre>";

print\_r($response);

echo "</pre>";

\*/

return ['code' => $httpcode, 'content' => $response]; // передача вызвавшей преграмме кода ошибки и массива ответа

} else {

print\_r('Ошибка авторизации <br>');

return false;

}

};

**Поиск записи по запросу**

function searchquery($token, $dbId, $fld, $query)

//$token - токен авторизации

//$dbId - ID базы данных

//$fld - область поиска,

//$query - запрос поиска

{

require('config.php');

$query = urlencode($query); // URL кодирование запроса

$request = $URL\_API.$REG\_DB."/".$dbId.$REG\_REC.$fld.$SPACE.$query; //формирование URL запроса (запрос GET)

$ch = curl\_init();// инициализация запроса cURL и формирование его полей

curl\_setopt($ch, CURLOPT\_URL, $request); //URL запроса

curl\_setopt($ch, CURLOPT\_HTTPHEADER, array($RESP\_CONT\_JSON, 'authorization: Bearer ' . $token)); // заголовок запроса

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true); //возврат кода результатов передачи

$result = json\_decode(curl\_exec($ch), true); //выполнение запроса и декодирование

$httpcode = curl\_getinfo($ch, CURLINFO\_HTTP\_CODE); // выделение кода ответа сервера

if ($httpcode == 200) {

print\_r('Код ответа на запрос поиска = '.$httpcode.'<br>');

/\*

echo "<pre>";

print\_r($result);

echo "</pre>";

\*/

return ['code' => $httpcode, 'content' => $result]; //передача вызвавшей программе кода ошибки и массива ответа

} else {

print\_r('Ошибка запроса <br>');

return false;

}

}

**Поиск записи по ID**

function searchRecord($token, $dbId, $libid)

//$token - токен авторизации,

//$dbId - ID базы данных,

//$libid - ID записи

{

require('config.php');

$request\_URL = $URL\_API.$REG\_DB."/".$dbId."/records/".$libid;//формирование URL запроса (запрос GET)

$ch = curl\_init();// инициализация запроса cURL и формирование его полей

curl\_setopt($ch, CURLOPT\_URL, $request\_URL); //URL запроса

curl\_setopt($ch, CURLOPT\_HTTPHEADER, array($RESP\_CONT\_JSON, 'authorization: Bearer ' . $token)); // заголовок запроса

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);//возврат кода результатов передачи

$result = json\_decode(curl\_exec($ch), true); //выполнение запроса и декодирование

$httpcode = curl\_getinfo($ch, CURLINFO\_RESPONSE\_CODE); // выделение кода ответа сервера

// print\_r('Запрос = '.$result['links']['self'].'<br>'); -- дополнительная отладочная информация

/\*

echo "<pre>";

print\_r($search);

echo "</pre>";

\*/

if ($httpcode === 200) {

print\_r('Код ответа на запрос = '.$httpcode.'<br>');

return ['code' => $httpcode, 'content' => $result];//передача вызвавшей преграмме кода ошибки и массива ответа

} else {

print\_r('Ошибка.<br>');

return false;

}

} // ---END OF searchquery

**Проверка качества JSON**

function isJSON($string){ // проверка качества сформированого JSON

$error = is\_string($string) && is\_array(json\_decode($string, true)) && (json\_last\_error() == JSON\_ERROR\_NONE) ? true : false;

print\_r(json\_last\_error\_msg());

return $error;

} // ---END OF isJSON

Для проверки валидности строки формата JSON, полученной в результате перекодирования ассоциативного массива пользовательского запроса (см. request.php – шаблоны запросов)

**Изменение записи в БД по ID**

function writeRecord($token, $dbId, $libid,$data\_string)

$token - токен авторизации

$dbId - ID базы данных

$libid - ID записи

$data\_string - новое поле записи или запись целиком

{

require('config.php');

$request\_URL = $URL\_API.$REG\_DB."/".$dbId."/records/".$libid; //формирование URL запроса (запросы типа PATCH ли PUT)

$data\_string = json\_encode($data\_string,JSON\_UNESCAPED\_UNICODE); // перекодирование ассоциативного массива запроса в JSON

if(isJSON($data\_string)) {echo " JSON Valid!<br>";}else{echo " JSON Bad-bad-bad!<br>";} // проверка качества JSON запроса

$ch = curl\_init(); // инициализация запроса cURL и формирование его полей

curl\_setopt($ch, CURLOPT\_URL, $request\_URL); //URL запроса

curl\_setopt($ch, CURLOPT\_CUSTOMREQUEST, "PATCH"); // тип запроса PUT или PATCH

curl\_setopt($ch, CURLOPT\_POSTFIELDS, $data\_string); //формирование текста запроса

curl\_setopt($ch, CURLOPT\_HTTPHEADER, array( 'Content-Type: application/json','Accept: application/vnd.api+json','authorization: Bearer ' . $token));

// заголовок запроса

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);//возврат кода результатов передачи

$write = json\_decode(curl\_exec($ch)); //выполнение запроса и декодирование

$httpcode = curl\_getinfo($ch, CURLINFO\_RESPONSE\_CODE); // выделение кода ответа сервера

if ($httpcode === 200) {

print\_r('Код запроса создания записи = '.$httpcode.'<br>');

return ['code' => $httpcode, 'content' => $write];

} else {

print\_r('Ошибка.<br>');

return false;

}

} // ---END OF writequery

**Дополнительные используемые файлы**

**// config.php – константы для формирования запроса**

$IDB = '425';

$FLD = 'TI';

$TODAY = date("d.m.Y");

$FILTER\_IP = '/([0-3]\d)(-|\/|\.|\\\)([0,1]\d)\2(20[0-2]\d)/';

$URL\_API = 'http://192.168.1.44/api/v1';

$REG\_AUTH = '/oauth2/token';

$REG\_DB = '/databases';

$REG\_REC = '/records?filter[query]=';

$SPACE = "%20";

//$LIBID = 'RSLA%255CBIBL%255C0000214413';

$LIBID = 'RSLA%255CBIBL%255C0000474018';

$REG\_CONT\_URL = 'Content-Type: application/x-www-form-urlencoded';

$REG\_CONT\_JSON = 'Content-Type: application/json';

$RESP\_CONT\_JSON = 'Accept: application/vnd.api+json';

**// grant.php – «секретные» слова для авторизации**

$GRANT\_TYPE = 'password';

$CLIENT\_ID = '354FE540-6100-436F-A212-7B29C4D09545';

$CLIENT\_SECRET = 'rhBQCWiIufQRooTtXcH';

$USERNAME = 'PRGUS';

$PASSWORD = 'wsDCrf7';

$SCOPE = 'read/write/admin';

**// request.php – шаблоны запросов**

// массив $request – для добавления целой запись в БД вместо существующей;

// массив $request\_field - добавление, удаление, изменение ТОЛЬКО поля в существующей записи БД

$request = [

"data" => [

"type" => "marcrecord",

"attributes" => [

"leader" => "00000nai0 2200000 i 450 ",

"fields" => [

[

"tag" => "100",

"ind1" => " ",

"ind2" => " ",

"subfields" => [

[

"code" => "a",

"data" => "20210219d1997 u y0rusy0189 ca"

]

]

],

[

"tag" => "101",

"ind1" => "0",

"ind2" => " ",

"subfields" => [

[

"code" => "a",

"data" => "rus"

]

]

],

[

"tag" => "102",

"ind1" => " ",

"ind2" => " ",

"subfields" => [

[

"code" => "a",

"data" => "RU"

]

]

],

[

"tag" => "106",

"ind1" => " ",

"ind2" => " ",

"subfields" => [

[

"code" => "a",

"data" => "s"

]

]

],

[

"tag" => "200",

"ind1" => "1",

"ind2" => " ",

"subfields" => [

[

"code" => "a",

"data" => "Северные крепости"

]

]

],

[

"tag" => "230",

"ind1" => " ",

"ind2" => " ",

"subfields" => [

[

"code" => "a",

"data" => "сетевой ресурс"

],

[

"code" => "6",

"data" => "тестовая запись UI API"

]

]

],

[

"tag" => "300",

"ind1" => " ",

"ind2" => " ",

"subfields" => [

[

"code" => "a",

"data" => "Дата обращения к ресурсу 02.10.2013"

]

]

],

[

"tag" => "801",

"ind1" => " ",

"ind2" => "0",

"subfields" => [

[

"code" => "a",

"data" => "RU"

],

[

"code" => "b",

"data" => "RGBI"

],

[

"code" => "c",

"data" => "20210219"

],

[

"code" => "g",

"data" => "rcr"

]

]

],

[

"tag" => "801",

"ind1" => " ",

"ind2" => "1",

"subfields" => [

[

"code" => "a",

"data" => "RU"

],

[

"code" => "b",

"data" => "RGBI"

],

[

"code" => "c",

"data" => "19980713"

]

]

],

[

"tag" => "856",

"ind1" => "4",

"ind2" => "0",

"subfields" => [

[

"code" => "u",

"data" => "http =>//www.nortfort.ru/"

]

]

],

[

"tag" => "909",

"ind1" => " ",

"ind2" => " ",

"subfields" => [

[

"code" => "a",

"data" => "Сетевой ресурс"

]

]

]

]

]

]

];

// add info in field of records

$record\_field = [

"data" => [

[

"op" => "replace", // add,remove,replace

"type" => "marcrecord",

"attributes" => [

"fields" => [

[

"tag" => "300",

"ind1" => " ",

"ind2" => " ",

"subfields" => [

[

"code" =>"a",

"data" => "Дата обращения к ресурсу 20.02.2021"

]

]

]

]

]

]

]

];

?>

/\*------ START OF RECORD DELETE

$ch = curl\_init();

curl\_setopt($ch, CURLOPT\_URL, $url);

curl\_setopt($ch, CURLOPT\_CUSTOMREQUEST, "DELETE");

curl\_setopt($ch, CURLOPT\_POSTFIELDS,$data\_json);

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);

$response = curl\_exec($ch);

curl\_close($ch);

------ END OF RECORD DELETE

\*/

?>