API Documentation

**#### Add a Book**

<?php

// http://localhost/API\_ITEC111/BREAD/add/addBook.php

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: POST");

header("Access-Control-Allow-Headers: Content-Type");

require '../../inc/dbh.inc.php';

function validateInput($data) {

$errors = [];

foreach ($data as $key => $value) {

if ($key !== 'bookId' && $key !== 'status' && empty($value)) {

$errors[] = $key;

}

}

return $errors;

}

function insertBook($data) {

global $conn;

try {

$sql = "INSERT INTO books (title, author, genre, publishedDate, pageCount, status)

VALUES (:title, :author, :genre, :publishedDate, :pageCount, :status)";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':title', $data['title']);

$stmt->bindParam(':author', $data['author']);

$stmt->bindParam(':genre', $data['genre']);

$stmt->bindParam(':publishedDate', $data['publishedDate']);

$stmt->bindParam(':pageCount', $data['pageCount']);

if (isset($data['status']) && $data['status'] !== null) {

$stmt->bindParam(':status', $data['status']);

} else {

$defaultStatus = 0;

$stmt->bindParam(':status', $defaultStatus);

}

$result = $stmt->execute();

if ($result) {

$matchingRowCount = countMatchingRows($data);

if ($matchingRowCount > 0) {

echo "You Have ".$matchingRowCount." same Book exist in the database";

return "A matching row already exists in the database.";

} else {

return "You inserted a new Book.";

}

} else {

return "Error adding the book.";

}

} catch (PDOException $e) {

return "Error: " . $e->getMessage();

}

}

function countMatchingRows($data) {

global $conn;

$selectSql = "SELECT COUNT(\*) as count FROM books WHERE

title = :title AND

author = :author AND

genre = :genre AND

publishedDate = :publishedDate AND

pageCount = :pageCount;";

$selectStmt = $conn->prepare($selectSql);

$selectStmt->bindParam(':title', $data['title']);

$selectStmt->bindParam(':author', $data['author']);

$selectStmt->bindParam(':genre', $data['genre']);

$selectStmt->bindParam(':publishedDate', $data['publishedDate']);

$selectStmt->bindParam(':pageCount', $data['pageCount']);

$selectStmt->execute();

$result = $selectStmt->fetch(PDO::FETCH\_ASSOC);

return $result['count'];

}

function getBookCount() {

global $conn;

$countSql = "SELECT COUNT(\*) as bookCount FROM books ";

$countStmt = $conn->prepare($countSql);

$countStmt->execute();

return $countStmt->fetch(PDO::FETCH\_ASSOC)["bookCount"];

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$data = [

'title' => $\_POST["title"] ?? null,

'author' => $\_POST["author"] ?? null,

'genre' => $\_POST["genre"] ?? null,

'publishedDate' => $\_POST["publishedDate"] ?? null,

'pageCount' => $\_POST["pageCount"] ?? null,

'status' => $\_POST["status"] ?? null,

];

$errors = validateInput($data);

if (!empty($errors)) {

echo "Error: The following fields are required: " . implode(", ", $errors);

exit();

}

if (insertBook($data)) {

$bookCount = getBookCount();

echo "'\nNew book added successfully. Total books now: $bookCount";

} else {

echo "Error adding the book.";

}

$conn = null;

} else {

echo "Invalid request method";

}

?>

**#### Explanation**

Endpoint: `POST /API\_ITEC111/BREAD/add/addBook.php`

Description: Add a new book to the database.

- Request:

- Method: `POST`

- Headers:

- `Content-Type: application/x-www-form-urlencoded`

- Parameters:

- `title` (string, required): Title of the book.

- `author` (string, required): Author of the book.

- `genre` (string, required): Genre of the book.

- `publishedDate` (string, required): Published date of the book (YYYY-MM-DD).

- `pageCount` (int, required): Number of pages in the book.

- `status` (int, optional): Status of the book (default is 0).

- Response:

- Success:

- Status Code: 200 OK

- Content: `You inserted a new Book.` or `A matching row already exists in the database.` or `You Have {matchingRowCount} same Book exist in the database`.

- Failure:

- Status Code: 400 Bad Request or 500 Internal Server Error

- Content: `Error: {error\_message}`

**#### Browse All Books**

<?php

// localhost/API\_ITEC111/BREAD/browse/fetchAllBooks.php

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: GET");

header("Content-Type: application/json");

require '../../inc/dbh.inc.php';

if ($\_SERVER["REQUEST\_METHOD"] == "GET") {

try {

$sql = "SELECT \* FROM books WHERE status = 0";

$stmt = $conn->prepare($sql);

$stmt->execute();

$result = $stmt->fetchAll(PDO::FETCH\_ASSOC);

if (count($result) > 0) {

$response = array(

"books" => $result,

"total\_books" => count($result)

);

echo json\_encode($response);

} else {

echo json\_encode(array("books" => [], "total\_books" => 0));

}

} catch (PDOException $e) {

echo json\_encode(array("error" => $e->getMessage()));

}

$stmt = null;

$conn = null;

} else {

echo json\_encode(array("error" => "Invalid request method"));

}

?>  
**#### Explanation**

- Endpoint: `GET /API\_ITEC111/BREAD/browse/fetchAllBooks.php`

- Description: Retrieve a list of all books in the database.

- Request:

- Method: `GET`

- Headers:

- `Content-Type: application/json`

- Response:

- Success:

- Status Code: 200 OK

- Content: JSON array of books and total\_books count.

- Failure:

- Status Code: 500 Internal Server Error

- Content: `{"error": "{error\_message}"}`

---

**#### Archive All Books**

<?php

//localhost/API\_ITEC111/BREAD/delete/archiveBooks.php

function(id)

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: DELETE");

header("Access-Control-Allow-Headers: Content-Type");

require '../../inc/dbh.inc.php';

if ($\_SERVER["REQUEST\_METHOD"] == "DELETE") {

try {

$sql = "UPDATE books SET status = 0";

$stmt = $conn->prepare($sql);

if ($stmt->execute()) {

echo json\_encode(array("message" => "All books are archived successfully"));

http\_response\_code(200); // 200 OK

} else {

echo json\_encode(array("error" => "Failed to archive books"));

http\_response\_code(500); // 500 Internal Server Error

}

} catch (PDOException $e) {

echo json\_encode(array("error" => "Database error: " . $e->getMessage()));

http\_response\_code(500); // 500 Internal Server Error

}

$stmt = null;

$conn = null;

} else {

echo json\_encode(array("error" => "Invalid request method"));

http\_response\_code(405); // 405 Method Not Allowed

}

?>

**#### Explanation**

- Endpoint: `DELETE /API\_ITEC111/BREAD/delete/archiveBooks.php`

- Description: Archive (set status to 0) all books in the database.

- Request:

- Method: `DELETE`

- Headers:

- `Content-Type: application/json`

- Response:

- Success:

- Status Code: 200 OK

- Content: `{"message": "All books are archived successfully"}`

- Failure:

- Status Code: 500 Internal Server Error

- Content: `{"error": "Failed to archive books"}`

---

**#### Delete a Book**

<?php

//localhost/API\_ITEC111/BREAD/delete/deleteBook.php?book\_id=1

//localhost/API\_ITEC111/BREAD/delete/deleteBook.php/{id}

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: DELETE");

header("Access-Control-Allow-Headers: Content-Type");

require '../../inc/dbh.inc.php';

if ($\_SERVER["REQUEST\_METHOD"] == 'DELETE') {

$bookId = isset($\_GET["book\_id"]) ? $\_GET["book\_id"] : null;

$title = isset($\_GET["title"]) ? $\_GET["title"] : null;

if (($bookId !== null && !is\_numeric($bookId)) || ($title !== null && empty($title))) {

echo "Error: Invalid input data format.";

http\_response\_code(400);

exit();

}

try {

if ($bookId !== null) {

$sql = "UPDATE books SET status = 1 WHERE bookId = :bookId";

} elseif ($title !== null) {

$sql = "UPDATE books SET status = 1 WHERE title = :title";

}

$stmt = $conn->prepare($sql);

if ($bookId !== null) {

$stmt->bindParam(':bookId', $bookId);

} elseif ($title !== null) {

$stmt->bindParam(':title', $title);

}

if ($stmt->execute()) {

echo "Book status updated successfully";

} else {

echo "Error: Unable to update book status. Please try again later.";

http\_response\_code(500);

}

} catch (PDOException $e) {

echo "Error: " . $e->getMessage();

http\_response\_code(500);

}

$stmt = null;

$conn = null;

} else {

echo "Invalid request method";

http\_response\_code(405);

}

?>

**#### Explanation**

- Endpoint: `DELETE /API\_ITEC111/BREAD/delete/deleteBook.php`

- Description: Delete a book by either `book\_id` or `title`.

- Request:

- Method: `DELETE`

- Headers:

- `Content-Type: application/json`

- Parameters:

- `book\_id` (int, optional): ID of the book to delete.

- `title` (string, optional): Title of the book to delete.

- Response:

- Success:

- Status Code: 200 OK

- Content: `"Book status updated successfully"`

- Failure:

- Status Code: 400 Bad Request or 500 Internal Server Error

- Content: `{"error": "Invalid input data format."}` or `{"error": "Unable to update book status."}`

**#### Update a Book**

<?php

// http://localhost/API\_ITEC111/CRUD/update/updateBase.php

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: PUT");

header("Access-Control-Allow-Headers: Content-Type");

require '../../inc/dbh.inc.php';

if ($\_SERVER["REQUEST\_METHOD"] == "PUT") {

$input\_data = file\_get\_contents("php://input");

$data = json\_decode($input\_data, true);

if ((isset($data['title']) || isset($data['bookId'])) && isset($data['fieldToUpdate']) && isset($data['newValue'])) {

$title = isset($data['title']) ? $data['title'] : null;

$bookId = isset($data['bookId']) ? $data['bookId'] : null;

$fieldToUpdate = $data['fieldToUpdate'];

$newValue = $data['newValue'];

if ((empty($title) && empty($bookId)) || empty($fieldToUpdate) || empty($newValue)) {

echo "Error: Invalid input data format.";

http\_response\_code(400);

exit();

}

try {

$sql = "UPDATE books SET $fieldToUpdate = :newValue WHERE title = :title OR bookId = :bookId";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':title', $title);

$stmt->bindParam(':bookId', $bookId);

$stmt->bindParam(':newValue', $newValue);

if ($stmt->execute()) {

echo "Record updated successfully";

} else {

echo "Error: Unable to update record. Please try again later.";

http\_response\_code(500);

}

} catch (PDOException $e) {

echo "Error: " . $e->getMessage();

http\_response\_code(500);

}

$stmt = null;

$conn = null;

} else {

echo "Error: Invalid input data 'title' or 'bookId', 'fieldToUpdate', and 'newValue' parameters are required.";

http\_response\_code(400);

}

} else {

echo "Invalid request method";

http\_response\_code(405);

}

?>

<?php

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: PUT");

header("Access-Control-Allow-Headers: Content-Type");

require '../../inc/dbh.inc.php';

$input\_data = file\_get\_contents("php://input");

$data = json\_decode($input\_data, true);

$bookId = isset($data['bookId']) ? $data['bookId'] : null;

$title = isset($data['title']) ? $data['title'] : null;

$author = isset($data['author']) ? $data['author'] : null;

$genre = isset($data['genre']) ? $data['genre'] : null;

$publishedDate = isset($data['publishedDate']) ? $data['publishedDate'] : null;

$ISBN = isset($data['ISBN']) ? $data['ISBN'] : null;

$pageCount = isset($data['pageCount']) ? $data['pageCount'] : null;

$status = isset($data['status']) ? $data['status'] : null;

if (empty($bookId) || empty($title) || empty($author) || empty($genre) || empty($publishedDate) || empty($pageCount) || empty($status)) {

echo "Error: All fields are required.";

http\_response\_code(400);

exit();

}

// Validate other input formats...

try {

$sql = "UPDATE books

SET title = :title,

author = :author,

genre = :genre,

publishedDate = :publishedDate,

ISBN = :ISBN,

pageCount = :pageCount,

status = :status

WHERE bookId = :bookId";

$stmt = $conn->prepare($sql);

$stmt->bindParam(':bookId', $bookId);

$stmt->bindParam(':title', $title);

$stmt->bindParam(':author', $author);

$stmt->bindParam(':genre', $genre);

$stmt->bindParam(':publishedDate', $publishedDate);

$stmt->bindParam(':ISBN', $ISBN);

$stmt->bindParam(':pageCount', $pageCount);

$stmt->bindParam(':status', $status);

if ($stmt->execute()) {

echo "Record updated successfully";

} else {

echo "Error: Unable to update record. Please try again later.";

http\_response\_code(500);

}

} catch (PDOException $e) {

echo "Error: " . $e->getMessage();

http\_response\_code(500);

}

$stmt = null;

$conn = null;

?>

**#### Explanation**

- Endpoint: `PUT /API\_ITEC111/CRUD/update/updateBase.php`

- Description: Update a book's information.

- Request:

- Method: `PUT`

- Headers:

- `Content-Type: application/json`

- Body:

```json

{

"bookId": 123,

"title": "Updated Title",

"author": "Updated Author",

"genre": "Updated Genre",

"publishedDate": "2023-01-01",

"ISBN": "1234567890123",

"pageCount": 300,

"status": 1

}

- Response:

- Success:

- Status Code: 200 OK

- Content: `"Record updated successfully"`

- Failure:

- Status Code: 400 Bad Request or 500 Internal Server Error

- Content: `{"error": "Invalid input data format."}` or `{"error": "Unable to update record."}`

**#### Search Books**

<?php

//localhost/API\_ITEC111/BREAD/read/searchBooks.php?search=search\_query

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: GET");

header("Content-Type: application/json");

require '../../inc/dbh.inc.php';

if ($\_SERVER["REQUEST\_METHOD"] == "GET") {

if (isset($\_GET["search"]) && !empty($\_GET["search"])) {

$searchInfo = $\_GET["search"];

try {

$sql = "SELECT \* FROM books WHERE

bookId LIKE :searchInfo OR

title LIKE :searchInfo OR

author LIKE :searchInfo OR

genre LIKE :searchInfo OR

publishedDate LIKE :searchInfo OR

ISBN LIKE :searchInfo OR

pageCount LIKE :searchInfo OR

status LIKE :searchInfo";

$stmt = $conn->prepare($sql);

$searchParam = "%" . $searchInfo . "%";

$stmt->bindParam(':searchInfo', $searchParam, PDO::PARAM\_STR);

$stmt->execute();

$result = $stmt->fetchAll(PDO::FETCH\_ASSOC);

echo json\_encode($result);

} catch (PDOException $e) {

echo json\_encode(array("error" => $e->getMessage()));

}

$stmt = null;

$conn = null;

} else {

echo json\_encode(array("error" => "Please specify, the data to search"));

}

} else {

echo json\_encode(array("error" => "Invalid request method"));

}

?>

**#### Explanation**

- Endpoint: `GET /API\_ITEC111/BREAD/read/searchBooks.php`

- Description: Search for books based on a search query.

- Request:

- Method: `GET`

- Headers:

- `Content-Type: application/json`

- Parameters:

- `search` (string, required): Search query.

- Response:

- Success:

- Status Code: 200 OK

- Content: JSON array of matching books.

- Failure:

- Status Code: 400 Bad Request or 500 Internal Server Error

- Content: `{"error": "{error\_message}"}`

---

**#### Retrieve Books by Column and Data**

<?php

// localhost/API\_ITEC111/BREAD/read/retrieveBase.php?book=column\_name&data=search\_value

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: GET");

header("Content-Type: application/json");

require '../../inc/dbh.inc.php';

if ($\_SERVER["REQUEST\_METHOD"] == "GET") {

$col = isset($\_GET["book"]) ? htmlspecialchars($\_GET["book"]) : null;

$data = isset($\_GET["data"]) ? htmlspecialchars($\_GET["data"]) : null;

if ($col && $data) {

try {

$sql = "SELECT \* FROM books WHERE $col LIKE :data";

$stmt = $conn->prepare($sql);

$searchParam = "%" . $data . "%";

$stmt->bindParam(':data', $searchParam, PDO::PARAM\_STR);

$stmt->execute();

$result = $stmt->fetchAll(PDO::FETCH\_ASSOC);

echo json\_encode($result);

} catch (PDOException $e) {

echo json\_encode(array("error" => $e->getMessage()));

}

$stmt = null;

$conn = null;

} else {

echo json\_encode(array("error" => "Invalid input parameters"));

}

} else {

echo json\_encode(array("error" => "Invalid request method"));

}

?>

**#### Explanation**

- Endpoint: `GET /API\_ITEC111/BREAD/read/retrieveBase.php`

- Description: Retrieve books based on a specified column and data.

- Request:

- Method: `GET`

- Headers:

- `Content-Type: application/json`

- Parameters:

- `book` (string, required): Column name to search.

- `data` (string, required): Search value.

- Response:

- Success:

- Status Code: 200 OK

- Content: JSON array of matching books.

- Failure:

- Status Code: 400 Bad Request or 500 Internal Server Error

- Content: `{"error": "Invalid input parameters"}` or `{"error": "{error\_message}"}`

**### Database Connection**

<?php

$host = "localhost";

$username = "root";

$password = "";

$dbname = "library\_db";

try {

$conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

echo "Connection failed: " . $e->getMessage();

exit();

}

?>

**#### Explanation**

- Connection Details:

- Host: `localhost`

- Username: `root`

- Password: `""` (empty string)

- Database: `library\_db`