

Project 1 Tutorial Slides

Joshua Sparks, Northern Kentucky University

Prepared for: Prof. Samuel Cho, ASE 230, Fall 2025

Overview:

The goal of this project was to create a Jira-like ticketing system that accomplishes the following tasks:

- Implement at least 8 rest APIs with at least 2 bearer tokens
- Allow for the creation of users, boards, issues, and comments
- Allow for the query of users, boards, issues, and comments by ID or all
- Allow for the authorized creation/update of board and issue information

Database Tables Breakdown:

```
+-----+
| Tables_in_jira_lite |
+-----+
| boards               |
| comments             |
| issues               |
| users                |
+-----+
```

get_user API

```
function get_user($conn, $id) {  
    $sql = "SELECT * FROM users WHERE id = ?";  
    $stmt = $conn->prepare($sql);  
    $stmt->bind_param("i", $id);  
    $stmt->execute();  
    $result = $stmt->get_result();  
  
    if ($result->num_rows > 0) {  
        $row = $result->fetch_assoc();  
        echo "Found user: " . $row["id"] . " (Name: " . $row["name"] . ") <br>";  
    } else {  
        echo "No User with ID $id<br>";  
    }  
    $stmt->close();  
}
```

Explanation for get_user API:

- `SELECT * FROM users WHERE id = ?`
 - selects all data from the users table where ID is the specified ID
- `$stmt->bind_param("i", $id)`
 - Binds the specified ID to the query
- `$stmt->execute()`
 - Executes the query against the database
- `$result = $stmt->get_result()`
 - Returns the results of the query
- `if ($result->num_rows > 0)`
 - Checks if any data was returned
 - Yes -> the user's ID and name is displayed
 - No -> displays "No user with ID \$id" message

Examples:

cURL Request:

```
curl http://localhost:8000/users.php/2
```

cURL Response:

```
Found user: 2 (Name: Test)
```

Examples cont.:

HTML/JavaScript Request:

```
document.querySelector('form').addEventListener('submit' , async function (event) {
    event.preventDefault();

    const form = event.target;
    const formData = new FormData(form);
    const userID = formData.get('userID');

    try {const response = await fetch(`../api_code/users.php/${userID}`);
        const data = await response.text();
        document.getElementById('result2').textContent =
            data;
    } catch (error) {document.getElementById('result2').textContent =
        'Error: ' + error.message;
    }});
```

HTML/JavaScript Response:

```
Found 1 users:<br>ID: 2 – Name: Test
```

get_all_users API

```
$sql = "SELECT * FROM users";  
$stmt = $conn->prepare($sql);  
$stmt->execute();  
$result = $stmt->get_result();  
  
if ($result->num_rows > 0) {  
    echo "Found " . $result->num_rows . " users:<br>";  
    while($row = $result->fetch_assoc()) {  
        echo "ID: " . $row["id"] . " - Name: " . $row["name"] . "<br>";  
    }  
} else {  
    echo "No users found<br>";  
}  
$stmt->close();
```


Explanation for get_all_users API:

- `SELECT * FROM users`
 - selects all data from the users table
- `$stmt = conn->prepare(sql);`
 - Prepares the query for the database
- `$stmt->execute()`
 - Executes the query against the database
- `$result = $stmt->get_result()`
 - Returns the results of the query
- `if ($result->num_rows > 0)`
 - Checks if any data was returned
 - Yes -> All user ID and name information is displayed
 - No -> displays "No users found" message

Examples:

cURL Request:

```
curl http://localhost:8000/users.php
```

cURL Response:

```
Found 1 users:<br>ID: 2 - Name: Test
```

Examples cont.:

HTML/JavaScript Request:

```
async function getUsersInfo() {
  try {
    const response = await fetch('../api_code/users.php');
    const data = await response.text();
    document.getElementById('result').textContent = data;
  } catch (error) {
    document.getElementById('result').textContent =
      'Error: ' + error.message;
  }
}
function getUsers() {
  getUsersInfo();
}
```

HTML/JavaScript Response:

```
Found 1 users:<br>ID: 2 - Name: Test
```

create_user API

```
$name = $_POST['name'] ?? '';
$id = $_POST['id'] ?? '';

$sql = "INSERT INTO users (name, id) VALUES (?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("si", $name, $id);

if ($stmt->execute()) {
    echo "User '$name' added successfully<br>";
} else {
    echo "Error: " . $stmt->error . "<br>";
}
$stmt->close();
```

Explanation for create_user API:

- `$name = $_POST['name'] ?? ''; & $id = $_POST['id'] ?? '';`
 - Records the user entered name and ID
- `$sql = "INSERT INTO users (name, id) VALUES (?, ?)";`
 - Insert the previously recorded data into the users table
- `$stmt = conn->prepare($sql);`
 - Prepares the query for the database
- `$stmt->bind_param("si", $name, $id);`
 - Binds the specified name and ID to the query
- `if ($stmt->execute())`
 - Checks if user creation is successful
 - Yes -> displays "User '\$name' added successfully"
 - No -> displays ""Error: " . \$stmt->error" message

Examples:

cURL Request:

```
curl -s -X POST -H "Content-Type: application/json" \ -d '{"name": "Test2","id": 1}' \ http://localhost:8000/users.php
```

cURL Response (no error message is good):

```
joshsparks@MacBookAir API Code %
```

Examples cont.:

HTML/JavaScript Request:

```
document.querySelector('#form2').addEventListener('submit' , async function (event) {event.preventDefault();

    const form = event.target;
    const formData = new FormData(form);

    try {const response = await fetch(`../api_code/users.php/`, {
        method: 'POST',
        body: formData});
        const data = await response.text();
        document.getElementById('result3').textContent =
            data;
    } catch (error) {
        document.getElementById('result3').textContent =
            'Error: ' + error.message;
    });
});
```

HTML/JavaScript Response:

```
User 'test' added successfully
```

get_board API

```
function get_board($conn, $id) {  
    $sql = "SELECT * FROM boards  
    WHERE id = ?";  
    $stmt = $conn->prepare($sql);  
    $stmt->bind_param("i", $id);  
    $stmt->execute();  
    $result = $stmt->get_result();  
  
    if ($result->num_rows > 0) {  
        $row = $result->fetch_assoc();  
        echo "Found board: " . $row["id"] . " (Name: " . $row["name"] . ") <br>";  
    } else {  
        echo "No Board with ID $id<br>";  
    }  
    $stmt->close();  
}
```


Explanation for get_board API:

- `SELECT * FROM boards WHERE id = ?`
 - selects all data from the boards table where ID is the specified ID
- `$stmt = conn->prepare(sql);`
 - Prepares the query for the database
- `$stmt->bind_param("i", $id)`
 - Binds the specified ID to the query
- `$stmt->execute()`
 - Executes the query against the database
- `$result = $stmt->get_result()`
 - Returns the results of the query
- `if ($result->num_rows > 0)`
 - Checks if any data was returned
 - Yes -> the board's ID and name is displayed

Examples:

cURL Request:

```
curl http://localhost:8000/boards.php/1
```

cURL Response:

```
Found board: 1 (Name: Test)
```

Examples cont.:

HTML/JavaScript Request:

```
document.querySelector('form').addEventListener('submit' , async function (event) {
    event.preventDefault();

    const form = event.target;
    const formData = new FormData(form);
    const boardID = formData.get('boardID');

    try {const response = await fetch(`../api_code/boards.php/${boardID}`);
        const data = await response.text();
        document.getElementById('result2').textContent =
            data;
    } catch (error) {
        document.getElementById('result2').textContent =
            'Error: ' + error.message; }});
```

HTML/JavaScript Response:

Found board: 1 (Name: Test)

get_all_boards API

```
$sql = "SELECT * FROM boards";  
$stmt = $conn->prepare($sql);  
$stmt->execute();  
$result = $stmt->get_result();  
  
if ($result->num_rows > 0) {  
    echo "Found " . $result->num_rows . " boards:<br>";  
    while($row = $result->fetch_assoc()) {  
        echo "ID: " . $row["id"]. " - Name: " . $row["name"]. "<br>";  
    }  
} else {  
    echo "No Boards found<br>";  
}  
$stmt->close();
```

Explanation for get_all_boards API:

- `SELECT * FROM boards`
 - selects all data from the boards table
- `$stmt = conn->prepare(sql);`
 - Prepares the query for the database
- `$stmt->execute()`
 - Executes the query against the database
- `$result = $stmt->get_result()`
 - Returns the results of the query
- `if ($result->num_rows > 0)`
 - Checks if any data was returned
 - Yes -> All board ID and name information is displayed
 - No -> displays "No boards found" message

Examples:

cURL Request:

```
curl http://localhost:8000/boards.php
```

cURL Response:

```
Found 1 boards:<br>ID: 1 - Name: Test:
```

Examples cont.:

HTML/JavaScript Request:

```
async function getBoardsInfo() {
  try {
    const response = await fetch('../api_code/boards.php');
    const data = await response.text();
    document.getElementById('result').textContent =
      data;
  } catch (error) {
    document.getElementById('result').textContent =
      'Error: ' + error.message;
  }}
function getBoards() {
  getBoardsInfo();
}
```

HTML/JavaScript Response:

```
Found 1 boards:<br>ID: 1 – Name: Test<br>
```

create_board API

```
$name = $_POST['name'] ?? '';
$id = $_POST['id'] ?? '';

$sql = "INSERT INTO boards (name, id) VALUES (?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("si", $name, $id);

if ($stmt->execute()) {
    echo "Board '$name' added successfully<br>";
} else {
    echo "Error: " . $stmt->error . "<br>";
}
$stmt->close();
```


Explanation for create_board API:

- `$name = $_POST['name'] ?? ''; & $id = $_POST['id'] ?? '';`
 - Records the user entered name and ID
- `$sql = "INSERT INTO boards (name, id) VALUES (?, ?)";`
 - Insert the previously recorded data into the boards table
- `$stmt = conn->prepare($sql);`
 - Prepares the query for the database
- `$stmt->bind_param("si", $name, $id);`
 - Binds the specified name and ID to the query
- `if ($stmt->execute())`
 - Checks if board creation is successful
 - Yes -> displays "Board '\$name' added successfully"
 - No -> displays ""Error: " . \$stmt->error" message

Examples:

cURL Request:

```
curl -s -X POST -H "Content-Type: application/json" \ -d '{"name": "Test2","id": 1}' \ http://localhost:8000/boards.php
```

cURL Response (no error message is good):

```
joshsparks@MacBookAir API Code %
```

Examples cont.:

HTML/JavaScript Request:

```
document.querySelector('#form2').addEventListener('submit' , async function (event) {event.preventDefault();

  const form = event.target;
  const formData = new FormData(form);

  try {const response = await fetch(`../api_code/boards.php/`, {
    method: 'POST',
    body: formData});
    const data = await response.text();
    document.getElementById('result3').textContent =
      data;
  } catch (error) {
    document.getElementById('result3').textContent =
      'Error: ' + error.message;}});
```

HTML/JavaScript Response (no error, bearer token working properly):

```
{"error":"Valid bearer token required"}
```

get_issue API

```
$sql = "SELECT * FROM issues WHERE id = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("i", $id);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    echo "Found issue: " . $row["id"] . " (Name: " . $row["name"] . ", Status: " . $row["status"] . ", Board: " . $row["board_id"] . ")<br>";
} else {
    echo "No Issue found with ID $id<br>";
}
$stmt->close();
```

Explanation for get_issue API:

- `SELECT * FROM issues WHERE id = ?`
 - selects all data from the issues table where ID is the specified ID
- `$stmt = conn->prepare(sql);`
 - Prepares the query for the database
- `$stmt->bind_param("i", $id)`
 - Binds the specified ID to the query
- `$stmt->execute()`
 - Executes the query against the database
- `$result = $stmt->get_result()`
 - Returns the results of the query
- `if ($result->num_rows > 0)`
 - Checks if any data was returned
 - Yes -> the issue's ID name status and associated board ID is displayed

Examples:

cURL Request:

```
curl http://localhost:8000/issues.php/3
```

cURL Response:

```
Found issue: 3 (Name: TestIssue, Status: Open, Board: 1)
```

Examples cont.:

HTML/JavaScript Request:

```
document.querySelector('form').addEventListener('submit' , async function (event) {event.preventDefault();

    const form = event.target;
    const formData = new FormData(form);
    const issueID = formData.get('issueID');

    try {const response = await fetch(`../api_code/issues.php/${issueID}`);
        const data = await response.text();
        document.getElementById('result2').textContent =
            data;
    } catch (error) {
        document.getElementById('result2').textContent =
            'Error: ' + error.message;}}});
```

HTML/JavaScript Response:

```
Found issue: 3 (Name: TestIssue, Status: Open, Board: 1)<br>
```

get_all_issues API

```
$sql = "SELECT * FROM issues";
$stmt = $conn->prepare($sql);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
echo "Found " . $result->num_rows . " issues:<br>";
while($row = $result->fetch_assoc()) {
    echo "ID: " . $row["id"]. " - Name: " . $row["name"].
        " - Board: " . $row["board_id"].
        " - Status: " . $row["status"]. "<br>";
}
} else {
    echo "No Issues found<br>";
}
$stmt->close();
```


Explanation for get_all_issues API:

- `SELECT * FROM issues`
 - selects all data from the issues table
- `$stmt = conn->prepare(sql);`
 - Prepares the query for the database
- `$stmt->execute()`
 - Executes the query against the database
- `$result = $stmt->get_result()`
 - Returns the results of the query
- `if ($result->num_rows > 0)`
 - Checks if any data was returned
 - Yes -> All issue information is displayed
 - No -> displays "No Issues found" message

Examples:

cURL Request:

```
curl http://localhost:8000/issues.php
```

cURL Response:

```
Found 1 issues:<br>ID: 3 – Name: TestIssue – Board: 1 – Status: Open<br>%
```

Examples cont.:

HTML/JavaScript Request:

```
async function getIssuesInfo() {
  try {
    const response = await fetch('../api_code/issues.php');
    const data = await response.text();
    document.getElementById('result').textContent =
      data;
  } catch (error) {
    document.getElementById('result').textContent =
      'Error: ' + error.message;
  }
}
function getIssues() {
  getIssuesInfo();}
```

HTML/JavaScript Response:

```
Found 1 issues:<br>ID: 3 – Name: TestIssue – Board: 1 – Status: Open<br>
```

create_issue API

```
$name = $_POST['name'] ?? '';
$board_id = $_POST['board_id'] ?? '';
$status = $_POST['status'] ?? '';

$sql = "INSERT INTO issues (name, board_id, status) VALUES (?, ?, ?)";
$stmt = $conn->prepare($sql);
$stmt->bind_param("sis", $name, $board_id, $status);

if ($stmt->execute()) {
    echo "Issue '$name' added successfully<br>";
} else {
    echo "Error: " . $stmt->error . "<br>";
}
$stmt->close();
```

Explanation for create_issue API:

- `$name = $_POST['name'] ?? ''; & $board_id = $_POST['board_id'] ?? '';`
 - Records the user entered name and ID
- `$sql = "INSERT INTO issues (name, board_id, status) VALUES (?, ?, ?)";`
 - Insert the previously recorded data into the issues table
- `$stmt = conn->prepare($sql);`
 - Prepares the query for the database
- `$stmt->bind_param("sis", $name, $board_id, $status);`
 - Binds the specified name, board ID, and status to the query
- `if ($stmt->execute())`
 - Checks if issue creation is successful
 - Yes -> displays "Issue '\$name' added successfully"
 - No -> displays ""Error: " . \$stmt->error" message

Examples:

cURL Request:

```
curl -s -X POST -H "Content-Type: application/json" \ -d '{"name": "TestIssue2","board_id": 2,"status": "Closed"}' \ http://localhost:8000/issues.php
```

cURL Response (no error message is good):

```
joshsparks@MacBookAir API Code %
```

Examples cont.:

HTML/JavaScript Request:

```
document.querySelector('#form2').addEventListener('submit' , async function (event) {
    event.preventDefault();

    const form = event.target;
    const formData = new FormData(form);

    try {const response = await fetch(`../api_code/issues.php/`, {
        method: 'POST',
        body: formData});
        const data = await response.text();
        document.getElementById('result3').textContent =
            data;
    } catch (error) {
        document.getElementById('result3').textContent =
            'Error: ' + error.message;}});
```

HTML/JavaScript Response (no error, bearer token working properly):

```
{"error": "Valid bearer token required"}
```

update_issue API

```
$raw_data = file_get_contents('php://input');  
$json_data = json_decode($raw_data, true);  
  
$new_name = $json_data['name'] ?? '';  
$new_board_id = $json_data['board_id'] ?? '';  
$new_status = $json_data['status'] ?? '';  
  
$sql = "UPDATE issues SET name = ?, board_id = ?, status = ? WHERE id = ?";  
$stmt = $conn->prepare($sql);  
$stmt->bind_param("sisi", $new_name, $new_board_id, $new_status, $id);  
  
if ($stmt->execute()) {  
    echo "Issue with ID $id updated successfully<br>";  
} else {  
    echo "Error updating issue: " . $stmt->error . "<br>";  
}  
$stmt->close();
```


Explanation for update_issue API:

- `$raw_data = file_get_contents('php://input'); json_data = json_decode(raw_data, true);`
 - Reads the raw input data and converts it into a PHP array
- `$new_name = $json_data['name'] ?? '';`
`$new_board_id = $json_data['board_id'] ?? '';`
`$new_status = $json_data['status'] ?? '';`
 - Records the user entered name, board ID, and status
- `$sql = "UPDATE issues SET name = ?, board_id = ?, status = ? WHERE id = ?";`
 - Insert the previously recorded updated data into the issues table
- `$stmt = conn -> prepare(sql);`
 - Prepares the query for the database
- `$stmt->bind_param("sisi", $new_name, $new_board_id, $new_status, $id);`
 - Binds the specified name, board ID, and status to the query

Examples:

cURL Request:

```
curl -s -X PATCH -H "Content-Type: application/json" \ -d '{"name":"TestIssue"}' \
http://localhost:8000/issues.php/3/name
```

cURL Response (no error, bearer token working properly):

```
{"error":"Valid bearer token required"}%
```

Examples cont.:

HTML/JavaScript Request:

```
document.querySelector('#form3').addEventListener('submit' , async function (event) {event.preventDefault();

  const form = event.target;
  const formData = new FormData(form);

  try {const response = await fetch(`../api_code/issues.php/`, {
    method: 'POST',
    body: formData});
    const data = await response.text();
    document.getElementById('result4').textContent =
      data;
  } catch (error) {
    document.getElementById('result4').textContent =
      'Error: ' + error.message;
  }});
```

HTML/JavaScript Response (no error, bearer token working properly):

```
{"error": "Valid bearer token required"}
```

update_issue_status API

```
$raw_data = file_get_contents('php://input');  
$json_data = json_decode($raw_data, true);  
$new_status = $json_data['status'] ?? '';  
  
$sql = "UPDATE issues SET status = ? WHERE id = ?";  
$stmt = $conn->prepare($sql);  
$stmt->bind_param("si", $new_status, $id);  
  
if ($stmt->execute()) {  
    echo "Issue with ID $id status updated successfully<br>";  
} else {  
    echo "Error updating issue: " . $stmt->error . "<br>";  
}  
$stmt->close();
```

Explanation for update_issue_status API:

- `$raw_data = file_get_contents('php://input'); json_data = json_decode(raw_data, true);`
 - Reads the raw input data and converts it into a PHP array
- `$new_status = $json_data['status'] ?? '';`
 - Records the user entered status
- `$sql = "UPDATE issues SET status = ? WHERE id = ?";`
 - Insert the previously recorded data into the issues table
- `$stmt = conn — > prepare(sql);`
 - Prepares the query for the database
- `$stmt->bind_param("si", $new_status, $id);`
 - Binds the specified status to the query
- `if ($stmt->execute())`
 - Checks if issue update is successful

Examples:

cURL Request:

```
curl -s -X PATCH -H "Content-Type: application/json" \ -d '{"status":"Open"}' \ http://localhost:8000/issues.php/3/status
```

cURL Response (no error, bearer token working properly):

```
{"error":"Valid bearer token required"}%
```

Examples cont.:

HTML/JavaScript Request:

```
document.querySelector('#form4').addEventListener('submit' , async function (event) {
    event.preventDefault();

    const form = event.target;
    const formData = new FormData(form);

    try {const response = await fetch(`../api_code/issues.php/`, {
        method: 'POST',
        body: formData});
        const data = await response.text();
        document.getElementById('result5').textContent =
            data;
    } catch (error) {
        document.getElementById('result5').textContent =
            'Error: ' + error.message;}});
```

HTML/JavaScript Response (no error, bearer token working properly):

```
{"error":"Valid bearer token required"}
```

add_issue_comment API

```
$comment = $_POST['comment'] ?? '';  
  
$sql = "UPDATE issues SET comment = ? WHERE id = ?";  
$stmt = $conn->prepare($sql);  
$stmt->bind_param("si", $comment, $id);  
  
if ($stmt->execute()) {  
    echo "Comment added to Issue $id successfully<br>";  
} else {  
    echo "Error: " . $stmt->error . "<br>";  
}  
$stmt->close();
```


Explanation for add_issue_comment API:

- `$comment = $_POST['comment'] ?? '';`
 - Records the user entered comment
- `$sql = "UPDATE issues SET comment = ? WHERE id = ?";`
 - Insert the previously recorded data into the issues table where ID is the specified ID
- `$stmt = conn->prepare($sql);`
 - Prepares the query for the database
- `$stmt->bind_param("si", $comment, $id);`
 - Binds the specified comment and ID to the query
- `if ($stmt->execute())`
 - Checks if comment creation is successful
 - Yes -> displays "Comment added to Issue \$id successfully"

Examples:

cURL Request:

```
curl -s -X POST -H "Content-Type: application/json" \ -d '{"comment": "Test comment"}' \ http://localhost:8000/issues.php/3/comments
```

cURL Response (no error message is good):

```
joshsparks@MacBookAir API Code %
```

Examples cont.:

HTML/JavaScript Request:

```
document.querySelector('#form5').addEventListener('submit' , async function (event) {
    event.preventDefault();

    const form = event.target;
    const formData = new FormData(form);

    try {const response = await fetch(`../api_code/issues.php/`, {
        method: 'POST',
        body: formData});
        const data = await response.text();
        document.getElementById('result6').textContent =
            data;
    } catch (error) {
        document.getElementById('result6').textContent =
            'Error: ' + error.message;}});
```

HTML/JavaScript Response (no error, bearer token working properly):

```
{"error": "Valid bearer token required"}
```

References:

Simple_PHP_Server_with_MYSQL.md, page number 9

Building a REST API Server with PHP.md, page numbers 13, 14, 15, 19, 31, 32, 33, 36

CRUD_operation.md, page numbers 1, 3, 5, 6, 8,

PUT and POST Requests.md, page numbers 32, 34, 35, 36, 38, 39

Bearer Token Authenticoitin.md, page numbers 11, 12, 14

Testing REST API Server.md, pages 25, 28, 29, 31, 33, 36

Frontend with JavaScript.md, pages 16, 19, 20, 21

Handle_User_Inputs_and_Formats.md, pages 3, 9

project1_outline.md marp format