

Creating a Database-Driven API with PHP and MySQL

In the previous tutorials, the API returned data from a JSON file.

In this tutorial, you will create an API that reads data from a MySQL database and returns it as JSON.

The System Architecture

The completed system will work like this:

```
security_events table
  ↓
api.php
  ↓
JSON
  ↓
JavaScript
```

Instead of reading a JSON file, the API will generate JSON directly from the database.

Create the API File

Create:

```
api_events.php
```

Add the database connection:

```
<?php

$conn = new mysqli(
    "localhost",
    "root",
    "",
    "project_db"
);

if ($conn->connect_error) {
    die("Connection failed: " .
        $conn->connect_error);
}

?>
```

Set the Content Type

Add:

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

The API will now return JSON instead of HTML.

Query the Database

Add:

```
$query = "
    SELECT *
    FROM security_events
";
```

```
$result =  
    $conn->query($query);
```

This retrieves all records from the table.

Create an Array

Add:

```
$events = [];
```

This array will hold all database records.

Loop Through the Results

Add:

```
while (  
    $row =  
    $result->fetch_assoc()  
) {  
  
    $events[] = $row;  
  
}
```

This converts each database record into an array item.

Convert the Array to JSON

Add:

```
echo json_encode(  
    $events,
```

```
        JSON_PRETTY_PRINT
    );
```

The completed API should now return JSON.

Test the API

Open:

```
http://localhost/api-demo/api_events.php
```

You should see:

```
[
  {
    "event_id": "1",
    "device_id": "CAM-01",
    "event_type": "motion_detected",
    "severity": "high"
  },
  {
    "event_id": "2",
    "device_id": "CAM-01",
    "event_type": "remote_access_attempt",
    "severity": "critical"
  }
]
```

The exact records will depend on your database.

Complete API File

```
<?php

$conn = new mysqli(
    "localhost",
    "root",
```

```
        "",
        "project_db"
    );

    if ($conn->connect_error) {
        die("Connection failed: " .
            $conn->connect_error);
    }

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

    $query = "
        SELECT *
        FROM security_events
    ";

    $result =
        $conn->query($query);

    $events = [];

    while (
        $row =
            $result->fetch_assoc()
    ) {

        $events[] = $row;

    }

    echo json_encode(
        $events,
        JSON_PRETTY_PRINT
    );

?>
```

Create a Webpage to Read the API

Create:

events.html

Add:

```
<!DOCTYPE html>
<html>
<head>
  <title>Security Events</title>
</head>
<body>

<h1>Security Events</h1>

<div id="output"></div>

<script>

fetch("api_events.php")
  .then(response => response.json())
  .then(data => {

    let html = `
      <table border="1">

        <tr>
          <th>Device</th>
          <th>Event Type</th>
          <th>Severity</th>
        </tr>
      `;

    data.forEach(event => {

      html += `
        <tr>
```

```
        <td>${event.device_id}</td>
        <td>${event.event_type}</td>
        <td>${event.severity}</td>
    </tr>
    `;

    });

    html += "</table>";

    document.getElementById(
        "output"
    ).innerHTML = html;

    });

</script>

</body>
</html>
```

Test the Complete System

Open:

```
http://localhost/api-demo/events.html
```

The page should display the records stored in MySQL.

Why This Is Useful

Previously:

```
JSON File
```

```
↓
```

```
JavaScript
```

Now:

MySQL Database

↓

PHP API

↓

JavaScript

This means:

- New records can be added to the database
- The API updates automatically
- The webpage updates automatically
- No JSON file needs to be edited manually

This is the foundation of many modern web applications.

Next tutorial: **Creating API Endpoints for Specific Data (Critical Events, Devices and Statistics).**

Revision #1

Created 2026-06-10 02:02:11 UTC by Mr Napper

Updated 2026-06-10 02:02:16 UTC by Mr Napper