

Creating API Endpoints for Specific Data

Focused endpoints return the smallest useful result for a particular user need. This tutorial adds filtered and summary endpoints using PDO and fictional records.

Endpoint for one severity

Create `api/events-by-severity.php`:

```
<?php
declare(strict_types=1);

require __DIR__ . '/../includes/database.php';

header('Content-Type: application/json; charset=utf-8');
header('Cache-Control: no-store');

$allowed = ['low', 'medium', 'high', 'critical'];
$severity = strtolower(trim($_GET['severity'] ?? ''));

if (!in_array($severity, $allowed, true)) {
    http_response_code(400);
    echo json_encode([
        'ok' => false,
        'error' => 'Choose low, medium, high or critical.'
    ]);
    exit;
}

$stmt = $pdo->prepare(
    'SELECT
        event_id,
        event_timestamp,
```

```

        device_id,
        room,
        event_type,
        severity
    FROM security_events
    WHERE severity = :severity
    ORDER BY event_timestamp DESC
    LIMIT 100'
);

$statement->execute(['severity' => $severity]);
$events = $statement->fetchAll();

echo json_encode([
    'ok' => true,
    'filter' => ['severity' => $severity],
    'count' => count($events),
    'events' => $events
]);

```

Test it with a fictional category:

```
http://localhost/project/api/events-by-severity.php?severity=high
```

The allow-list rejects unexpected values and the prepared statement keeps the supplied value separate from the SQL.

Endpoint for summary totals

Create `api/event-summary.php`:

```

<?php
declare(strict_types=1);

require __DIR__ . '/../includes/database.php';

header('Content-Type: application/json; charset=utf-8');
header('Cache-Control: no-store');

```

```

$statement = $pdo->query(
    'SELECT severity, COUNT(*) AS total
    FROM security_events
    GROUP BY severity
    ORDER BY severity'
);

$rows = $statement->fetchAll();

$summary = [];

foreach ($rows as $row) {
    $summary[] = [
        'severity' => $row['severity'],
        'total' => (int) $row['total']
    ];
}

echo json_encode([
    'ok' => true,
    'summary' => $summary
]);

```

This endpoint returns processed information rather than every event record. It can support an accessible table or chart.

Choose endpoints from user needs

Useful endpoints might answer:

- Which fictional events require urgent review?
- How many events occurred in each category?
- Which events belong to a selected device?
- How has the event count changed over time?

Avoid creating many endpoints that expose the same unnecessary fields.

Check

- Invalid filters return HTTP 400 and a clear JSON error.
 - User-supplied values are validated and passed through prepared statements.
 - Totals are cast to numbers before encoding.
 - Responses contain only the required fields.
 - No personal information or credentials are exposed.
 - All demonstration records are fictional.
-

Revision #3

Created 2026-06-10 02:02:28 UTC by Mr Napper

Updated 2026-08-19 23:03:45 UTC by Mr Napper