

API Project for the "Roller Coaster System"

General Assumptions

- The system manages roller coasters and their assigned wagons.
- The system provides information on whether roller coasters have a sufficient number of wagons and staff.
- The system operates in two modes: development and production, managed via Docker Compose.
- Data related to coasters and wagons is stored in a Redis server.

API for the Roller Coaster System

1. Registering a New Roller Coaster

- Endpoint: POST /api/coasters
- Adds a new roller coaster to the system, including details such as:
 - Number of staff
 - Daily customer capacity
 - Track length (in meters)
 - Operating hours
- Example Data:
{ "staff_count": 16, "daily_customers": 60000, "track_length": 1800, "operating_hours_from": "08:00", "operating_hours_to": "18:00" }

2. Registering a New Wagon

- Endpoint: POST /api/coasters/:coasterId/wagons
- Adds a new wagon to a specific roller coaster with seat capacity and speed data.
- Example Data: { "seat_count": 32, "wagon_speed": 1.2 }

3. Deleting a Wagon

- Endpoint: DELETE /api/coasters/:coasterId/wagons/:wagonId
- Removes a selected wagon from a roller coaster.

4. Updating a Roller Coaster

- Endpoint: PUT /api/coasters/:coasterId
- Updates existing roller coaster data, such as number of staff, daily customer capacity, and operating hours.
- Track length remains unchanged.

Development Mode

- Restricted access for external users.
- Logs all types of events.

- Development data must not interfere with production data.

Production Mode

- Logs only "warning" and "error" messages.
- Production data is completely separate from development data.

Redis - Queue and Wagon Management

- Roller coasters operate within specified hours.
- Wagons must complete their ride before the coaster's operating hours end.
- Wagons require a 5-minute break before restarting.

Staff Management

- 1 staff member per coaster, 2 additional staff per wagon.
- System alerts if staff is insufficient or excessive.

Customer Management

- The system tracks daily customers per roller coaster.
- If demand exceeds capacity, the system calculates missing wagons and staff.
- If the capacity is excessive (2x required), the system identifies excess wagons and staff.

Statistics & Monitoring (Console)

- Real-time monitoring service in PHP CLI (asynchronous).
- Uses Redis storage and ReactPHP/Swoole for async processing.
- Example Log: [2024-11-29 00:12:30] Roller Coaster A1 - Missing 2 staff, 2 wagons.

Technical Requirements

- PHP 8.0+ and CodeIgniter 4 framework.
- Deployment using PHP-FPM and Nginx.