

The background of the image is a dark, atmospheric scene of the Hogwarts castle silhouette against a sky filled with warm, golden and orange hues of a setting or rising sun. The castle's towers and spires are visible, with some windows showing light.

Hogwarts API Project

by Arnur

Hogwarts API

A Go language-based API that allows users to create, update, and view Hogwarts faculties and students. Inspired by a rewatch of the Harry Potter series, the project includes user authentication and authorization.

Features

- CRUD Operations:
 - Faculties: Create, read, update, delete. Pagination, Filtering, Sorting
 - Students: Create, read, update, delete. Pagination, Filtering, Sorting
- User Management:
 - User registration and activation.
 - User authentication and token generation.
- Endpoint Summary:
 - /v1/healthcheck - Health check endpoint.
 - /v1/faculties - Manage faculties.
 - /v1/students - Manage students.
 - /v1/users - Register and activate users.
 - /v1/tokens/authentication - User authentication.



Technology Stack:

- Language: Go
- Database: PostgreSQL
- Dependencies: [List any Go packages used]



• Project Uniqueness

Inspired by the Harry Potter series, bringing the magical world into a structured API format.

Showcases practical usage of Go for building RESTful APIs with user authentication and authorization.

Demonstrates how to manage complex relationships between entities like faculties and students.



Entities



```
CREATE TABLE IF NOT EXISTS faculties (
    id bigserial PRIMARY KEY,
    created_at timestamp with time zone NOT NULL
DEFAULT NOW(),
    founder text NOT NULL,
    title text NOT NULL,
    year integer NOT NULL,
    Runtime integer NOT NULL,
    version integer NOT NULL DEFAULT 1
);
CREATE TABLE IF NOT EXISTS students (
    id bigserial PRIMARY KEY,
    created_at timestamp with time zone NOT NULL
DEFAULT NOW(),
    name text NOT NULL,
    surname text NOT NULL,
    study_year integer NOT NULL,
    age integer NOT NULL,
    Faculty_id INTEGER NOT NULL,
    Runtime integer NOT NULL,
    version integer NOT NULL DEFAULT 1,
    FOREIGN KEY (Faculty_id) REFERENCES faculties(id)
);
```



Project Complexity

- Database Schema:
 - Faculties Table: Stores information about Hogwarts faculties.
 - Students Table: Stores information about students, with foreign key relationship to faculties.
 - User and Permissions Tables: Manages user roles and permissions.
- Authentication and Authorization:
 - Secure endpoints with user roles.
 - JWT-based token authentication.

- API Endpoints:
 - Comprehensive CRUD operations for faculties and students.
 - Secure endpoints with role-based access control.
- Pagination, Filtering, Sorting
 - For both of the entities there is an option to use pagination filtering and sorting, for example
 - http://localhost:4000/v1/faculties/1/students?page_size=2&page=1