

Api/Portal Exercise

Python / GCP/ AWS

MidLevel



1

Introducción

Instrucciones

- ▶ Crear un **API REST**, desplegada en la capa gratuita de **Google Cloud Platform** o **Amazon Web Services**, de acuerdo a las siguientes directrices.
- ▶ El código debe estar publicado en **Github**.
- ▶ Un audio explicando tu soluciones y decisiones tomadas, debe ser incluido.
- ▶ Tienes 4 días para enviar la solución.
- ▶ El API será probado con Postman, usando pruebas automatizadas.

2

Restricciones

Restricciones Técnicas

- ▶ Python 3.6+ must be used for the implementation
- ▶ La base de datos debe ser Mysql, y debe estar instalada en el mismo servidor donde la aplicación se ejecuta.
- ▶ El historial de Git debe reflejar la implementación del API Rest
- ▶ El código fuente y sus comentarios deben estar en inglés.
- ▶ Preferir la capa gratuita de GCP o AWS para el despliegue. Puedes destruir aquellos servidores después del proceso de reclutamiento.
- ▶ Incluir scripts de despliegue como parte del código fuente en Github.
- ▶ Escoger un framework MVC y un ORM para la solución, evitando así escribir consultas SQL en el código.

3

Problema

- ▶ Crear un API Rest para manejar 2 recursos: drugs (fármacos) y vaccinations (vacunaciones), de acuerdo a las siguientes definiciones:
 - ▷ Drug:
 - name
 - code (max 10 chars, único)
 - description (max 255 chars)
 - ▷ Vaccination
 - rut (rut válido con dígito verificador de la persona vacunada)
 - dose (volumen aplicado en cm3 $0.15 \leq \text{dose} \leq 1.0$)
 - date (fecha de vacunación)
 - drug
- ❑ El API debe ser JSON para las peticiones y respuestas de los endpoints.
- ❑ El API debe validar el JSON body del request, basándose en las restricciones mencionadas anteriormente.
- ❑ Una prueba unitaria debe ser implementada para probar la función de validación de rut.
- ❑ El API debe ser protegida usando token JWT.

4

API

REST API

- Usar los métodos, cabeceras y estado del protocolo HTTP para crear los endpoints:
 - Métodos:
 - GET leer información del recurso
 - POST crear recurso y ejecutar procesos.
 - PUT actualizar recursos.
 - DELETE eliminar recursos.
 - Cabeceras:
 - Content-Type
 - Authorization
 - Accept
 - Estado
 - 200 (OK) -> recurso leído o actualizado
 - 201 (Created) -> recurso creado
 - 204 (Not Content) -> recurso eliminado
 - 4xx (Client Errors)
 - 5xx (Server Errors)

4

API

REST API Spec - Drug

- ▶ **GET /drugs** -> Retorna arreglo con todos los fármacos
- ▶ **GET /drugs/:id** -> Retorna un fármaco, por su id. Retorna **status 404** si el fármaco no se encuentra.
- ▶ **POST /drug** -> Crea un fármaco basado en el json payload. Retorna **status 201** con un json body si hay éxito. Retorna **status 400** en error de validación o cuando la **cabecera "Content-Type"** no esté seteada como **"application/json"**. Retorna **status 500** en cualquier otro error.
- ▶ **PUT /drug/:id** -> Actualiza un fármaco, basado en su id y en el json payload. Retorna **status 200** si hay éxito. Retorna **status 404** si el fármaco no es encontrado. Retorna **status 400** en error de validación o cuando la **cabecera "Content-Type"** no esté seteada como **"application/json"**. Retorna **status 500** en cualquier otro error.
- ▶ **DELETE /drug/:id** -> Elimina un fármaco basado en su id. Retorna **status 204** si hay éxito. Retorna **status 404** si el fármaco no es encontrado. Si el fármaco tiene vacunaciones asociadas, enviará **status 400**.

4

API

REST API Spec - Vaccination

- ▶ **GET /vaccination** -> Retorna arreglo con todas las vacunaciones.
- ▶ **GET /vaccination/:id** -> Retorna una vacunación, por su id. Retorna **status 404** si la vacunación no se encuentra.
- ▶ **POST /vaccination** -> Crea una vacunación basado en el json payload. Retorna **status 201** con un json body si hay éxito. Retorna **status 400** en error de validación o cuando la **cabecera "Content-Type"** no esté seteada como **"application/json"**. Retorna **status 500** en cualquier otro error.
- ▶ **PUT /vaccination/:id** -> Actualiza una vacunación, basado en su id y en el json payload. Retorna **status 200** si hay éxito. Retorna **status 404** si la vacunación no es encontrada. Retorna **status 400** en error de validación o cuando la **cabecera "Content-Type"** no esté seteada como **"application/json"**. Retorna **status 500** en cualquier otro error.
- ▶ **DELETE /vaccination/:id** -> Elimina una vacunación basado en su id. Retorna **status 204** si hay éxito. Retorna **status 404** si el fármaco no es encontrado.

4

API

REST API Spec - JWT

- ▶ **GET /token** -> Obtiene un token JWT válido para acceder al API. No recibe ningún parámetro o contraseña.

5

Despliegue

Despliegue

- La aplicación de estar desplegada en GCP o AWS, usando el servicio de cómputo de tu elección (GKE, EKS, App Engine, GCE, Amazon EC2, etc)
- Se requiere usar la capa gratuita de GCP o AWS.
- Opcionalmente, si es posible configurar un despliegue automático, disparado por los cambios aplicados en la rama master de git.

6

Revisión

Proceso de revisión

- ▶ Seguir las instrucciones y entregar a tiempo será considerado como positivo
- ▶ El proceso de desarrollo será revisado evaluando el historial de commits.
- ▶ El audio explicativo es importante para asegurar que la persona que implementó el ejercicio entendió cómo resolverlo y las decisiones tomadas para llegar a cabo. El audio debe estar en español.
- ▶ Simpleza es importante
- ▶ Es importante de incluir scripts de despliegue para integración continua.