

Práctica 3: Creación de API REST con Bottle

Sistemas Distribuidos

Curso: 2019/2020

Índice

1. Partes de la práctica a entregar	2
2. Normas para la realización de la práctica	2
3. Parte 1: Servicio Web REST con Bottle	2
3.1. Mejoras	3
4. Parte 2: Cliente	3
5. Evaluación	4
6. FECHA LIMITE DE ENTREGA	4

Este documento indica los requisitos para la Práctica 3.

1. Partes de la práctica a entregar

1. Código fuente (.zip)

- Las prácticas se corregirán en Linux, para aquellos que utilicen un S.O. distinto (MacOS o Windows) se recomienda testear el código en una distribución Linux antes de entregarlo.

2. Memoria descriptiva del trabajo realizado (pdf)

- Se debe proporcionar una descripción detallada del programa implementado así como de su funcionamiento y ejecución.
- Cualquier mejora implementada pero no descrita en la memoria no será evaluada.
- Si se utilizan librerías adicionales, debe indicarse en la memoria, así como la versión de Python utilizada.

2. Normas para la realización de la práctica

- Realización por parejas (se admiten trabajos individuales, pero no de tres estudiantes).
- Los trabajos deberán ser entregados obligatoriamente antes de la fecha de entrega fijada en la actividad habilitada en el campus virtual.
- Las dos partes del trabajo son obligatorias. Si una de ellas no se realiza el trabajo se considerará no presentado.

3. Parte 1: Servicio Web REST con Bottle

En esta práctica vamos a trabajar con el framework Bottle visto en el Seminario 3. El objetivo es crear un servicio web de gestión de habitaciones de un hotel.

El servicio debe poseer una lista de habitaciones y cada habitación poseerá, mínimo, los siguientes atributos:

- Identificador.
- Número de plazas.
- Lista de equipamiento (por ejemplo: armario, aire acondicionado, caja fuerte, escritorio, etc.).
- Ocupada (sí/no).

El servicio deberá implementar, mínimo, los siguientes **endpoints**:

1. Dar de alta una nueva habitación.

2. Modificar los datos de una habitación.
3. Consultar la lista completa de habitaciones.
4. Consultar una habitación mediante identificador.
5. Consultar la lista de habitaciones ocupadas o desocupadas.

Queda a **juicio del estudiante** identificar qué tipo de operaciones HTTP debe implementar cada **endpoint** (PUT, POST o GET).

Para los **endpoints** 2, 4 y 5, es **obligatorio** el uso de parámetros en el **path** de la URL.

Aunque hemos visto que REST permite transmitir los datos en cualquier formato, se recomienda utilizar JSON¹, tal y como se ha visto en el seminario.

Se valorará positivamente el manejo adecuados de los errores (parámetros incorrectos, **endpoint** no encontrado, operaciones no permitidas) en el servicio y su notificación al cliente.

3.1. Mejoras

- Crear un **endpoint** para eliminar una habitación mediante una operación HTTP DELETE.
- Hacer que la información que maneja el servicio sea persistente, almacenando la información de las habitaciones en un fichero. ^{2 3}
- Crear una clase “Room” para facilitar el manejo de la información por parte del servicio. ⁴
- Crear los **endpoints** adicionales que el estudiante considere oportuno.

4. Parte 2: Cliente

Crear un cliente que permita a un usuario interactuar con la API desarrollada.

Este cliente, le mostrará un menú al usuario con las posibles operaciones que se podrán realizar (de acuerdo a las implementadas en el servicio), solicitará los parámetros necesarios y realizará la llamada al servicio. Posteriormente, mostrará la información al usuario (ver Figura 1).

¹<https://realpython.com/python-json/>

²<https://www.pythonforbeginners.com/files/reading-and-writing-files-in-python>

³<https://stackabuse.com/reading-and-writing-json-to-a-file-in-python/>

⁴<https://docs.python.org/3/tutorial/classes.html>

```
Elige que opción deseas realizar:
  1. Dar de alta una nueva habitación
  2. Modificar los datos de una habitación
  3. Consultar la lista completa de habitaciones
  4. Consultar una habitación mediante identificador
  5. Consultar la lista de habitaciones ocupadas o desocupadas
  6. Salir

> 4

Introduce el identificador de la habitación:

> 80

Habitación 80:
- 3 plazas.
- Equipamiento: TV, Wifi y escritorio.
- Libre.

Elige que opción deseas realizar:
...
```

Figura 1: Ejemplo de funcionamiento del cliente

5. Evaluación

La nota de la práctica se evalúa de la siguiente forma:

- Parte 1: 70 %
- Parte 2: 30 %

Cada parte seguirá la siguiente rúbrica:

- El programa tiene errores sintácticos o semánticos: 0 puntos
- El programa tiene errores de ejecución: 3 puntos
- El programa funciona sin errores, pero hace lo mínimo y las **explicaciones** son muy escuetas: 5 puntos
- El programa funciona perfectamente, se ha realizado alguna de las mejoras propuestas: 7,5 puntos
- El programa funciona perfectamente, es avanzado, y se han realizado muchas de las mejoras propuestas: 10 puntos

6. FECHA LIMITE DE ENTREGA

La fecha de entrega y presentación será la especificada en el Campus Virtual.