Práctica de servicios Web (I): servidores

Alvaro Tapia Muñoz, Jose Luis Bueno Pachón, Carlos Marín Corbera, Carmen González Ortega, Altair Bueno Calvente

11 oct 2022

Contents

Introducción

Descripción de los servicios

A2ReservasREST

Este microservicio se encarga de proporcionar los datos sobre las reservas. Esta desarrollado en Python utilizando el framework FastAPI. Para la persistencia de datos utiliza MongoDB, a través del driver Motor.

Para facilitar el despliegue de la aplicación, se proporciona un fichero `Dockerfile`. Las instrucciones detalladas sobre como desplegar la aplicación se pueden encontrar en el fichero `README.md`, dentro de la carpeta del proyecto

Requisitos considerados

- **Al menos dos operaciones de consulta o búsqueda parametrizada**: `GET /booking`, `GET /booking/{booking_id}`, `DELETE /booking/{booking_id}`

Esquema de las entidades

Los documentos almacenados en Mongo mantienen la el siguiente esquema:

```
{
  // Clave primaria de Mongo
  "_id": "ObjectId",
  // Identificador del usuario que realiza la reserva
  "user_id": "string",
  // Identificador de la vivienda
  "house_id": "string",
  // Fecha de inicio de la reserva
  "start_date": "date(YYYY-MM-DD)",
  // Fecha de fin de la reserva
  "end_date": "date(YYYY-MM-DD)"
}
```

Endpoints REST disponibles

- `GET /booking`: Devuelve una lista de reservas que cumplan con los filtros especificados. 10 reservas como máximo. Los parámetros de consulta son:

  - `skip`: Número de reservas a ignorar
  - `user_id`: Identificador de usuario
  - `before_date`: Fechas de fin anteriores
  - `after_date`: Fechas de inicio posteriores
  - `sort_by`: Campo utilizado para ordenar
  - `ascending`: Ordenar de forma ascendente. Por defecto: `false`

- `POST /booking`: Crea una nueva reserva. El cuerpo de la petición es un json con los siguientes campos:

  - `user_id`: Identificador del usuario
  - `house_id`: Identificador de la vivienda
  - `start_date`: Inicio de la reserva
  - `end_date`: Fin de la reserva

- `GET /booking/{booking_id}`: Devuelve toda la información sobre la reserva con identificador `booking_id`
- `DELETE /booking/{booking_id}`: Elimina la reserva con identificador `booking_id`
- `GET /ping`: Ruta utilizada para validar que el servicio se encuentra disponible

La documentación completa para los endpoints REST desarrollados se puede encontrar en el propio servidor, bajo la ruta `/docs`. Además, se adjunta una copia local en el fichero `openapi.json`, dentro de la carpeta del proyecto.