

Plan de pruebas Restful Booker – David Enriquez

Objetivo: Implementar pruebas automatizadas para los servicios proporcionados por Restful Booker con el fin de identificar en etapas tempranas fallos en los servicios que se integran en la aplicación.

Definición de herramientas: Para la realización de las pruebas se debe implementar un plan de ejecución manual a través de la herramienta Postman y un plan de ejecución automatizado con el framework Karate.

Lenguaje de pruebas: Para la definición de los casos de prueba se deben escribir y estructurar bajo la metodología BDD que implementa el lenguaje Gherkin.

Alcance: Este plan de pruebas esta delimitado a los servicios de Auth, Booking y Ping, incluye escenarios de éxito y flujos alternos que se identifican como respuestas posibles en el consumo de los servicios. No se incluye pruebas sobre la interfaz de usuario.

Criterios de Aceptación: Todos los servicios desarrollados deben cumplir los siguientes criterios

- Para los escenarios de éxito debe obtener respuesta válida del servicio
- Se debe cumplir los contratos del servicio de acuerdo con el cuerpo de las peticiones y de las respuestas definidos
- Las respuestas de los servicios en el estado y el cuerpo deben ser acordes a los escenarios probados

Prerrequisitos:

- Servicios desplegados en el entorno de pruebas
- Documentación de los servicios a automatizar
- Datos de entrada válidos para la consulta de reservas
- Datos de entrada válidos para la creación de reservas

Casos de prueba para servicio de autenticación

Escenario: Autenticación exitosa

- Dado un usuario con credenciales válidas
- Cuando envía una petición para la creación de un token
- Entonces recibe una respuesta exitosa

Escenario: Autenticación fallida por usuario inválido

- Dado un usuario con username inválido
- Y con password válido
- Cuando envía una petición para la creación de un token
- Entonces recibe una respuesta fallida

Escenario: Autenticación fallida por contraseña inválida

- Dado un usuario con username válido
- Y con password inválido
- Cuando envía una petición para la creación de un token
- Entonces recibe una respuesta fallida

Casos de prueba para servicios de reservas

Escenario: Crear reserva exitosamente

- Dado un usuario que requiere crear una reserva nueva
- Cuando ingresa los datos completos y correctos de la reserva
- Entonces la reserva se crea exitosamente

Escenario: Error al crear la reserva por no enviar datos requeridos

- Dado un usuario que requiere crear una reserva nueva
- Cuando ingresa los datos incompletos de la reserva
- Entonces se genera error en la petición

Escenario: Eliminar una reserva exitosamente

- Dado un usuario con una reserva
- Y autenticación en el servicio válida
- Cuando se envía la petición para eliminar la reserva con un id vigente
- Entonces se confirma que la reserva ha sido eliminada exitosamente

Escenario: Eliminar una reserva con un ID que no existe

- Dado que se requiere eliminar una reserva
- Cuando se envía la petición con un ID que no existe
- Entonces se genera error en la petición

Escenario: Eliminar una reserva con autenticación inválida

- Dado un usuario con una reserva
- Y autenticación en el servicio inválida
- Cuando se envía la petición para eliminar la reserva

- Entonces se genera error por autenticación inválida

Escenario: Obtener información de una reserva exitosamente

- Dado un usuario que requiere consultar una reserva vigente
- Cuando realiza la petición de consulta con un ID válido
- Entonces se obtiene la información de la reserva
- Y se verifica que la respuesta contenga los datos y formato esperado

Escenario: Obtener información de una reserva con un ID que no existe

- Dado un usuario que requiere consultar una reserva
- Cuando realiza la petición de consulta con un ID inválido
- Entonces se obtiene se retorna un error en la consulta

Escenario: Consultar la lista de reservas creadas

- Dado un usuario que requiere consultar la lista de reservas realizadas
- Cuando realiza la petición de consulta
- Entonces se obtiene la lista de ID's de las reservas existentes

Escenario: Obtener información de reservas con un parámetro de consulta

- Dado un usuario que requiere consultar información de una reserva
- Y dispone de información específica de la reserva a consultar
- Cuando realiza la petición de consulta con el parámetro definido
- Entonces si se encuentran coincidencias con el parámetro de búsqueda se obtiene el ID asociado a las reservas coincidentes

Escenario: Obtener información de reservas con múltiples parámetros de consulta

- Dado un usuario que requiere consultar información de una reserva
- Y dispone de información específica de la reserva a consultar
- Cuando realiza la petición de consulta con los parámetros definidos
- Entonces si se encuentran coincidencias con los parámetros de búsqueda se obtiene el ID asociado a las reservas coincidentes

Escenario: Obtener información de reservas con parámetros de consulta vacíos o errados

- Dado un usuario que requiere consultar información de una reserva
- Y dispone de información específica de la reserva a consultar
- Cuando realiza la petición de consulta con parámetros vacíos o valores errados
- Entonces se genera error en la consulta

Escenario: Actualizar parcialmente la información de una reserva exitosamente

- Dado un usuario que requiere actualizar uno o varios de los parámetros de la reserva
- Y autenticación en el servicio válida
- Cuando realiza la petición de actualización con los parámetros definidos
- Entonces se genera la actualización exitosamente

Escenario: Actualizar parcialmente la información de una reserva con parámetros inválidos

- Dado un usuario que requiere actualizar uno o varios de los parámetros de la reserva
- Y autenticación en el servicio válida
- Cuando realiza la petición de actualización con parámetros errados o inválidos
- Entonces se genera error en la actualización

Escenario: Actualizar parcialmente la información de una reserva no existente

- Dado un usuario que requiere actualizar uno o varios de los parámetros de la reserva
- Y autenticación en el servicio válida
- Cuando realiza la petición de actualización con un ID que no existe
- Entonces se genera error en la actualización

Escenario: Actualizar parcialmente la información de una reserva con autenticación inválida

- Dado un usuario que requiere actualizar uno o varios de los parámetros de la reserva
- Y autenticación en el servicio es inválida
- Cuando realiza la petición de actualización
- Entonces se retorna error en la respuesta del servicio

Escenario: Actualizar todos los parámetros de la reserva exitosamente

- Dado un usuario que requiere actualizar todos los datos de la reserva
- Y autenticación en el servicio es válida
- Cuando realiza la petición de actualización
- Entonces se actualiza los datos de la reserva exitosamente

Escenario: Actualizar todos los parámetros de la reserva exitosamente

- Dado un usuario que requiere actualizar todos los datos de la reserva
- Y autenticación en el servicio es válida
- Cuando realiza la petición de actualización
- Entonces se actualiza los datos de la reserva exitosamente

Escenario: Actualizar todos los parámetros de una reserva que no existe

- Dado un usuario que requiere actualizar todos los datos de la reserva
- Y autenticación en el servicio es válida
- Cuando realiza la petición de actualización con un ID que no existe
- Entonces no se permite la actualización de los datos

Escenario: Actualizar todos los datos de una reserva con autenticación inválida

- Dado un usuario que requiere actualizar todos los parámetros de la reserva
- Y autenticación en el servicio es inválida
- Cuando realiza la petición de actualización
- Entonces se retorna error en la respuesta del servicio

Escenario: Actualizar todos los parámetros de una reserva con campos vacíos o inválidos

- Dado un usuario que requiere actualizar todos los datos de la reserva
- Y autenticación en el servicio es válida
- Cuando realiza la petición de actualización con campos requeridos vacíos o inválidos
- Entonces no se permite la actualización de los datos

Casos de prueba para servicios de ping

Escenario: Obtener el estado del servicio exitosamente

- Dado un usuario que requiere consultar el estado del servicio
- Cuando realiza una petición de tipo GET al servicio
- Entonces retorna un código de respuesta
- Y el código indica el estatus actual del servicio