

SERVICIOS WEB RESTFUL - BOOKER

Planes de Pruebas Automatizadas

Versión del documento: 0001

Fecha: 28/02/2023

| | |
|---|-------------------------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-------------------------------|

INFORMACIÓN GENERAL

| | | | |
|------------------------|------------------------|-------------------------|------------|
| Proyecto | Yape - Desafío Técnico | | |
| Entregable | Plan de Pruebas | | |
| Responsable | Andrés Felipe Gallego | | |
| Versión/Edición | 0001 | Fecha Versión | 28/02/1993 |
| Aprobado por | | Fecha Aprobación | DD/MM/AAAA |

REGISTRO DE CAMBIOS

| Versión doc | Causa del Cambio | Responsable del Cambio | Fecha del Cambio |
|--------------------|-------------------------|-------------------------------|-------------------------|
| 0001 | Creación del documento | Andrés Felipe Gallego | 28/02/1993 |
| | | | |
| | | | |
| | | | |

| | |
|---|-------------------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-------------------------|

ÍNDICE

1 INTRODUCCIÓN

1.1 Objeto

2 ALCANCE DE LAS PRUEBAS

3 RIESGOS

4 DEFINICIÓN DE LOS CASOS DE PRUEBAS

5 ESTRATEGIA DE EJECUCIÓN DE PRUEBAS

| | |
|---|-------------------------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-------------------------------|

1 INTRODUCCIÓN

1.1 Objeto

El objetivo de este documento es identificar y detallar las pruebas que van a ejecutarse para garantizar el correcto funcionamiento de los servicios correspondientes al API desarrollada para el proyecto.

Así mismo, contendrá el alcance de las pruebas, los riesgos, la definición de los casos de prueba y la estrategia de pruebas que se implementará para su ejecución.

2 ALCANCE DE LAS PRUEBAS

- No se realizarán pruebas E2E que requieran confirmar la información en bases de datos.
- No se realizarán pruebas a desarrollos web.
- Las pruebas se realizarán exclusivamente a servicios web.

Las pruebas se centrarán en el consumo de los servicios descritos a continuación:

| Funcionalidad | Servicios | Métodos http |
|---------------|----------------------|--------------|
| Auth | CreateToken | POST |
| Booking | GetBookingIds | GET |
| | GetBooking | GET |
| | CreateBooking | POST |
| | UpdateBooking | PUT |
| | PartialUpdateBooking | PATCH |
| | DeleteBooking | DELETE |
| Ping | HealthCheck | GET |

| | |
|---|-------------------------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-------------------------------|

3 RIESGOS

Durante el desarrollo de las pruebas pueden presentarse algunos riesgos que pueden afectar al plan de trabajo planteado en este documento. Así mismo puede afectar los tiempos de entrega por parte del QA Automatizador.

Los riesgos analizados y que se deben considerar son los siguientes:

- El software no hace lo que se espera.
- Tiempos de respuesta inadecuados.
- Inestabilidad en el ambiente de pruebas.
- Inestabilidad en los servicios.
- Falta de accesibilidad para el consumo de los servicios.
- Cambios en el alcance del proyecto.

Entre los impactos que podrían generar tenemos:

- Pérdida de confianza en el producto.
- Pérdida de clientes.
- Pérdida de dinero.
- Reprocesos.
- Incumplimiento con las fechas de entrega.

| | |
|---|-----------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-----------------|

4 DEFINICIÓN DE LOS CASOS DE PRUEBAS

Los casos de prueba a tener en cuenta para el proyecto serán documentados en lenguaje Gherkin. En el siguiente recuadro se asocian los casos de prueba que se automatizan.

| Tag caso de prueba: | @CreateAuthToken |
|---|------------------|
| Funcionalidad | Auth |
| Feature: creates a new auth token to use for access to the PUT and DELETE booking | |
| Prerrequisitos Permisos para consumir el servicio | |
| Scenario: create a new authorization token Given the user prepares the service When the user sends the credentials to create a new auth token Then the user will get the status code 200 | |

| Tag caso de prueba: | @HealthCheck |
|--|--------------|
| Funcionalidad | Ping |
| Feature: consume health check endpoint to confirm whether the API is up and running | |
| Prerrequisitos Permisos para consumir el servicio | |
| Scenario: confirm whether the api is up and running Given the user prepares the service When the user consumes the HealthCheck Then the user will get the status code 201 | |

| | |
|---|-------------------------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-------------------------------|

| | |
|---|-----------------------|
| Tag caso de prueba: | @CreateBooking |
| Funcionalidad | Booking |
| Feature: consumes booking api to consult, create, delete and update the bookings | |
| Prerrequisitos Permisos para consumir el servicio | |
| Background: Given the user prepares the service | |
| Scenario: create a booking | |
| When the user sends data to create a new booking firstname lastname totalprice depositpaid checkin checkout additionalneeds Jim Brown 674 true 2018-01-01 2019-01-01 Breakfast | |
| Then the user will get the status code 200 | |

| | |
|--|-----------------------|
| Tag caso de prueba: | @GetBookingIds |
| Funcionalidad | Booking |
| Feature: consumes booking api to consult, create, delete and update the bookings | |
| Prerrequisitos Permisos para consumir el servicio | |
| Background: Given the user prepares the service | |
| Scenario: get booking ids | |
| When the user consumes the service booking ids Then the user will get the status code 200 | |

| | |
|---|-----------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-----------------|

| |
|--|
| |
|--|

| | |
|---|-------------|
| Tag caso de prueba: | @GetBooking |
| Funcionalidad | Booking |
| Feature: consumes booking api to consult, create, delete and update the bookings | |
| Prerrequisitos Permisos para consumir el servicio | |
| Background: Given the user prepares the service | |
| Scenario: get a specific booking Given the user gets the existing booking id When the user consumes the service booking with specific id Then the user will get the status code 200 | |

| | |
|---|-----------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-----------------|

| | |
|---|----------------|
| Tag caso de prueba: | @UpdateBooking |
| Funcionalidad | Booking |
| Feature: consumes booking api to consult, create, delete and update the bookings | |
| Prerrequisitos Permisos para consumir el servicio | |
| Background: Given the user prepares the service | |
| Scenario: update a booking Given the user gets the existing booking id When the user sends data to update an existing booking firstname lastname totalprice depositpaid checkin checkout additionalneeds Andres Gallego 557 true 2023-03-02 2023-03-08 Breakfast Then the user will get the status code 200 | |

| | |
|---|-----------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-----------------|

| | |
|--|-----------------------|
| Tag caso de prueba: | @PartialUpdateBooking |
| Funcionalidad | Booking |
| Feature: consumes booking api to consult, create, delete and update the bookings | |
| Prerrequisitos Permisos para consumir el servicio | |
| Background: Given the user prepares the service Scenario: partial update booking Given the user gets the existing booking id When the user sends data to update partially an existing booking firstname lastname Camilo Mendez Then the user will get the status code 200 | |

| | |
|---|-------------------------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-------------------------------|

| | |
|---|-----------------------|
| Tag caso de prueba: | @DeleteBooking |
| Funcionalidad | Booking |
| Feature: consumes booking api to consult, create, delete and update the bookings | |
| Prerrequisitos Permisos para consumir el servicio | |
| Background: Given the user prepares the service | |
| Scenario: delete a booking | |
| Given the user gets the existing booking id When the user authorizes delete an existing booking Then the user will get the status code 201 | |

| | |
|---|-------------------------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-------------------------------|

4 ESTRATEGIA DE PRUEBAS

Se implementarán pruebas automatizadas haciendo uso de las siguientes herramientas:

Patrón de Diseño: Para las pruebas se implementará el patrón de diseño ScreenPlay con las siguientes capas.

main/java

- **Exceptions:** El manejo de excepciones personalizadas es una parte fundamental en el proyecto ya que brinda la posibilidad de tener un control sobre las fallas que se puedan presentar con un mensaje diciente.
Esta excepciones ayudan a detectar donde se ha presentado la falla y no tener re procesos a la hora de identificar el origen y motivo
- **Models:** Es la representación de los objetos que hacen parte de nuestra aplicación y están relacionados al negocio. Por ejemplo Usuario, Facturas
Son clases que representan un tipo de objeto de la vida real los cuales cuentan con unos atributos que los hace únicos y realizan acciones con un fin determinado.
- **Tasks:** esta capa se enfoca al negocio (alto nivel). Las Tareas (task) se componen por un conjunto de acciones o interacciones con el objetivo de plasmar lo que un Actor puede hacer.
- **Utils:** Pueden existir métodos para usar en la automatización que no forman parte del objeto de prueba pero que sin embargo se deben construir para así garantizar la funcionalidad de la misma.
Algunos de estos métodos pueden ser: lectura de archivos externos, acciones puntuales sobre tipos de datos como String o Enteros.

test/java

- **runners:** Los Runners son las clases ejecutables para los features en los cuales se encuentran los escenarios diseñados para nuestras pruebas.
- **stepdefinitions:** son la representación en código de los escenarios diseñados en los features.

test/resources

- **Features:** En esta carpeta se agregan los archivos que contienen los escenarios definidos y documentados con lenguaje Gherkin.
Para esto se implementará la herramienta Cucumber.

| | |
|---|-------------------------------|
| Planes de Pruebas Servicios Web RestFul - Booker | Versión 0001 |
|---|-------------------------------|

Lenguaje de programación: Java

Librerías/Dependencias:

- implementation group: 'net.serenity-bdd', name: 'serenity-core', version: '2.0.49'
- implementation group: 'net.serenity-bdd', name: 'serenity-junit', version: '2.0.49'
- implementation group: 'net.serenity-bdd', name: 'serenity-screenplay', version: '2.0.49'
- implementation group: 'net.serenity-bdd', name: 'serenity-cucumber', version: '1.9.31'
- implementation group: 'net.serenity-bdd', name: 'serenity-screenplay-webdriver', version: '2.0.49'
- testImplementation group: 'org.slf4j',name: 'slf4j-simple',version: '1.7.7'
- implementation group: 'net.serenity-bdd', name: 'serenity-screenplay-rest', version: '2.0.49'

Repositorio:

El código del proyecto se agregará al repositorio creado en GitHub

El cual contendrá adicionalmente la documentación necesaria y los insumos para las pruebas.

Ruta:

https://github.com/AndresGallego08/Yape_Automation.git