

Aplicaciones RESTful:

Documentación y Testing

Práctica

1) Documentación del servidor

Confeccionar la documentación de APIs correspondiente al TP-04. Esta debe incluir todos los puntos de acceso al programa, detallando el comportamiento esperado, tanto en sus casos de éxito, como en sus casos de fallo esperados.

2) Cliente de pruebas

Realizar un cliente de pruebas que pueda consumir los servicios publicados por nuestro servidor del TP-04.

3) Tests al servidor

Realizar un programa que, utilizando nuestro cliente de pruebas, pueda consumir y probar el funcionamiento del servidor desarrollado en el TP-04, verificando que cumpla con lo especificado en la documentación.

Metodología de trabajo

- Leer la documentación del servidor a testear.
- Por cada punto de acceso del servidor:
 - Pensar en los distintos escenarios en que podría usarse.
 - Armar un lote de pruebas con datos arbitrarios, y calcular qué respuesta debería recibir por parte del servidor, al enviarle cada uno de los datos de mi lote de prueba.
 - Para cada uno de los datos pensados, escribir una función que envíe ese dato al servidor y registre de alguna manera si la respuesta generada es la correcta.
 - Se deben cubrir todo el comportamiento descrito en la documentación, incluyendo recibir respuestas válidas ante peticiones válidas, como también recibir respuestas con estado de error ante peticiones inválidas o imposibles de concretarse.

Tener en cuenta que cada test debe ser independiente de los demás. Por este motivo, es posible que al comienzo del test deban realizarse una o más operaciones para establecer el escenario que se desea testear. De igual manera, al finalizar el test, puede que sea necesario realizar algún tipo de operación que deshaga lo hecho durante el mismo.

Trabajo adicional

Para que nuestros tests no se pisen entre sí, ni corran el riesgo de que datos de una prueba interfieran con consultas de otra prueba sobre nuestro servidor, nuestros tests deben correr en forma estrictamente secuencial. Como ya sabemos, las operaciones secuenciales pueden llegar a demorar mucho tiempo, y es por ese motivo que en aquellos escenarios en que es posible intentamos realizar nuestras operaciones *en paralelo*. Se te ocurre alguna forma de implementar las pruebas para que éstas se ejecuten todas al mismo tiempo, sin interferirse entre sí?