Engineer

Pruebas de contrato

pruebas de acuerdos de interfaz entre componentes

Qué es

Las pruebas de contrato comprueban que dos componentes coinciden en sus expectativas sobre las solicitudes, respuestas o mensajes que intercambian. Una pantalla de seguimiento de envíos y su servicio pueden superar sus pruebas por separado. Aun así, la pantalla fallará si busca un campo llamado `status` y el servicio envía `state` . Una prueba de la interfaz expresa qué necesita la pantalla y verifica que el servicio puede proporcionarlo.

Pact describe un método en el que el consumidor utiliza un sustituto controlado del proveedor. Su prueba registra las interacciones en un contrato. El proveedor verifica a continuación ese contrato mediante su implementación real en unas condiciones determinadas. Para los mensajes, la evidencia equivalente corresponde al mensaje que un lado produce y otro consume. Los dos pueden trabajar y publicar versiones de forma independiente, siempre que la evidencia identifique las versiones que en realidad se usarán juntas.

Otros equipos llaman prueba de contrato a verificar un proveedor a partir de una descripción publicada de la interfaz, como un documento OpenAPI. Sirve cuando esa especificación es el acuerdo compartido. Puede precisar campos, códigos de estado y valores permitidos. Los ejemplos guiados por el consumidor muestran además qué combinaciones utiliza un cliente y qué supuestos incorpora. El equipo se puede servir de los dos métodos. Una prueba a través de la integración real comprueba cuestiones adicionales, como el encaminamiento y la configuración del entorno.

Utilice el concepto en una decisión

Aplique «¿Qué son las pruebas de contrato?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. La parte que el usuario no ve: dónde se guardan los datos, qué reglas se aplican y cómo pueden comunicarse otros sistemas con ella.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Diseño y operación de 6 flujos ETL/ELT — 6 flujos ETL en tres motores de bases de datos

Diseño y operación de 6 flujos ETL/ELT Póngase en contacto →

Por qué importa

Una interfaz suele durar más que el código que la introdujo. El equipo del servicio puede renombrar un campo y cambiar sus pruebas sin actualizar un cliente que se mantiene aparte. Una aplicación móvil puede seguir usando una versión anterior durante meses. Incluso un cambio de tipo importa si determina el mensaje que aparecerá en pantalla. Un contrato verificable permite ver esas dependencias antes de publicar.

Con versiones independientes, interesa saber cuáles siguen siendo compatibles. Quizá un cliente antiguo solo comprende unos pocos estados y el servicio necesita incorporar uno intermedio. Los equipos pueden preparar primero al consumidor, aceptar las dos formas durante la transición y retirar una de ellas cuando dejen de usarla las versiones relevantes. El orden seguro depende de los contratos registrados, del uso real y de la política de publicación del sistema. El resultado favorable de una pareja reciente no cubre todos los clientes que siguen en uso.

Las pruebas deberían expresar lo que de verdad se acuerda. Si vale cualquier referencia con formato correcto, verifica el formato sin exigir el valor exacto del ejemplo. Si un campo determina qué ve la persona usuaria, comprueba el tipo y los valores que el cliente entiende. Un ejemplo bien escogido protege el comportamiento compartido y deja al proveedor margen para modificar los detalles internos.

Cómo se aplica en la práctica

Consideremos una aplicación ficticia de seguimiento de paquetes. Pide un envío por su referencia. Si el paquete existe y está en camino, la pantalla espera una respuesta de éxito, una referencia y un valor de texto que pueda convertir en una etiqueta. Si no se encuentra el paquete, espera una respuesta distinta y ofrece revisar la referencia. Cada intercambio parte de una condición, por ejemplo, que el envío solicitado se encuentre en tránsito o que no exista ningún registro con ese identificador.

Haz que el código real del consumidor formule la solicitud contra el sustituto controlado. Verifica que la aplicación muestra el resultado adecuado para cada ejemplo. Aplica después los intercambios a la ruta real de respuesta del proveedor. Prepara datos deterministas para el estado correspondiente y restablécelos entre pruebas. Si se sustituye la base de datos o la red del transportista, conserva en ejecución el análisis de la solicitud, la lógica del servicio y la generación de la respuesta que utiliza el consumidor.

Cambia provisionalmente el campo `status` a `state` sin modificar el cliente. La verificación debería señalar que falta el campo y nombrar el intercambio afectado. La presencia de un campo opcional que el consumidor ignora puede ser válida si el acuerdo lo permite. Vincula la evidencia a sus versiones. Mezclar un resultado del proveedor nuevo con una copia del contrato antiguo sin dejar constancia del origen impide saber qué combinación se verificó.

En nuestra nota sobre enlaces a aplicaciones dentro de billeteras móviles de Solana, varias integraciones esperan rutas y codificaciones distintas. La nota registra esas diferencias e incluye comprobaciones con dispositivos. Una comprobación de los enlaces generados podría proteger la forma acordada para cada billetera. Un ensayo real permite observar si el navegador entrega el enlace a la aplicación instalada y termina el recorrido. Cada prueba responde una pregunta distinta.

Dónde surgen los problemas

Un contrato se vuelve rígido si exige que coincidan todos los bytes de una respuesta. Puede ser válido añadir un campo sin uso o cambiar el formato de un mensaje auxiliar. Exige un valor, un orden o una unidad cuando el consumidor dependa de ellos o la especificación pública los establezca. Deja que los aspectos accidentales puedan variar.

Las pruebas también pierden valor si sortean la interfaz. Verificar un archivo de ejemplo escrito a mano puede dejar sin ejecutar el código que genera la respuesta. Comprobar que el sustituto devuelve un dato preparado tampoco demuestra que el consumidor pueda interpretarlo. Ejecuta el código real de la parte que estás comprobando y registra qué dependencias se han sustituido.

Por último, un único recorrido normal deja dudas sobre los datos ausentes, los identificadores desconocidos y los clientes antiguos. Localiza los errores que el cliente sabe presentar y las versiones que el proveedor se compromete a admitir.

Conserva pruebas de integración para ejercer el encaminamiento y la configuración. El contrato aporta una comprobación más rápida y enfocada que el equipo puede repetir ante cada cambio pertinente.

Lecturas adicionales

- Documentación de Pact sobre qué comprueban las pruebas de contrato
- Comprobación de consumidores y proveedores en Pact
- Estados del proveedor en Pact
- Especificación OpenAPI para describir interfaces HTTP

Glosario

Elija según sus restricciones reales

Describa el flujo de trabajo, las interfaces existentes y los usuarios que necesitan el cambio. Podemos definir una entrega comprobable y un traspaso que permita a su equipo mantener lo entregado.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué son las pruebas de contrato?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/contract-testing/