

API Testing

- Cada vez que se hace una llamada a un servidor, se realiza y cuenta como una solicitud de API. Las pruebas de API consisten en enviar solicitudes de API y comprobar sus respuestas. Existen métodos específicos utilizados para las pruebas de API.
- En el método HTTP, todas las llamadas tienen una respuesta y un código de estado. Los códigos de estado HTTP ayudan a identificar la causa del problema cuando una página web u otro recurso no se carga correctamente. Cuando el desarrollador define las API, identifica qué método HTTP se puede utilizar.

source: shutterStock

API Testing

¿Cómo es una prueba API?



API Testing

The Api URL

Es la solicitud HTTP para invocar una API concreta. Por ejemplo, API para la funcionalidad de inicio de sesión. Si la estructura de URL es Mobile/User/Login y la URL de su dominio es `http://domain.com`, entonces la URL de API para invocar la API desde el navegador/herramientas será:

`http://domain.com/Mobile/User/Login`

API Testing

HTTP Metodo

En el caso de las API RESTful, utilizan métodos HTTP para clasificar las API en función del tipo de llamada realizada al servidor. Por ejemplo, POST, GET, etc. El método POST envía datos al servidor, mientras que el método GET recupera los datos del servidor.

source: testLodge Blog

API Testing

Payload / Carga útil

Define la estructura o el modelo de los datos de entrada que se suministrarán a la API. Por ejemplo, si podemos utilizar la siguiente estructura de datos para la API de inicio de sesión:

```
{  
  username :string  
  password: string  
  device_id: string  
  token: string  
}
```

API Testing

Request Sample Data / Solicitar datos de muestra

Contiene los datos de entrada reales que se pasan a la API como carga útil. Puede tener tantos casos de prueba como desee basados en estos datos de muestra. Para el escenario de inicio de sesión, los datos de muestra podrían tener este aspecto:

```
{  
  "device_id": "F4325-C3423-32453",  
  "token" : "f5jkl54jkl54gj4oi23443",  
  "app_language" : "en"  
}
```

API Testing

Expected Result / Resultado esperado

Response Code / Código de respuesta

Representa el código de respuesta de la solicitud de API. 200 OK debería ser el código de respuesta de una petición API correcta. Hay otros códigos de respuesta como 400 Bad request, 401

Unauthorized, 403 Forbidden, 404 Not found, 500 Internal Server Error etc. Sería útil si pudiera anotar el código de respuesta de la API desde la consola de salida de la herramienta de prueba de API.

API Testing

Mensaje de resultado de la respuesta

Para cada entrada, puede haber diferentes mensajes de éxito de salida de la API. Debe averiguar los mensajes de respuesta correspondientes y documentarlos en sus casos de prueba. Algunos mensajes comunes para la API de inicio de sesión serán:

SUCCESS, INACTIVE_ACCOUNT,
INVALID_PASSWORD, USER_NOT_FOUND,
INVALID_DEVICE_ID, ERROR

API Testing

Respuesta Resultado Muestra

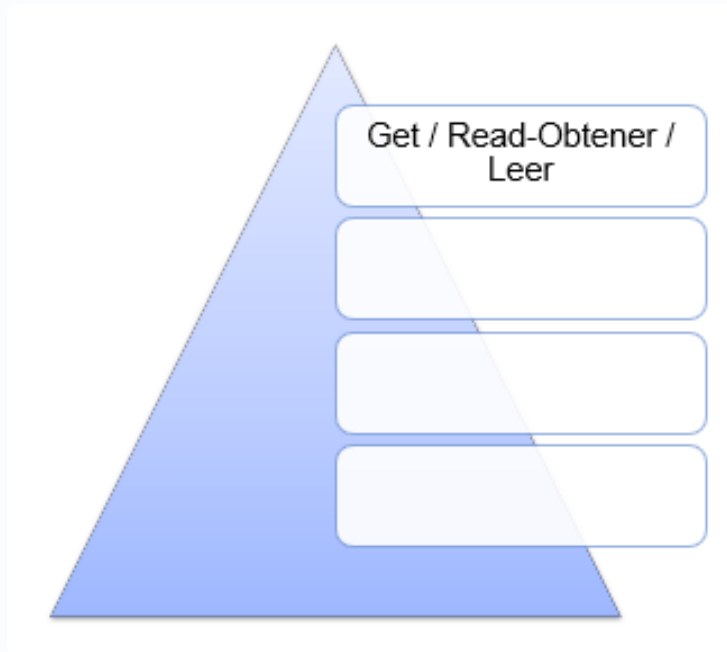
Estos son los datos de salida para cada una de las combinaciones de datos de entrada. Necesitará estos datos para poder validarlos con la salida real de los resultados de la API. Un ejemplo de una operación de inicio de sesión correcta es:

```
{  
  "result": "SUCCESS",  
  "data": {  
    "id": 7093,  
    "email": "[email protected]",  
    "created_at": "2019-04-01"  
  }  
}
```

API Testing

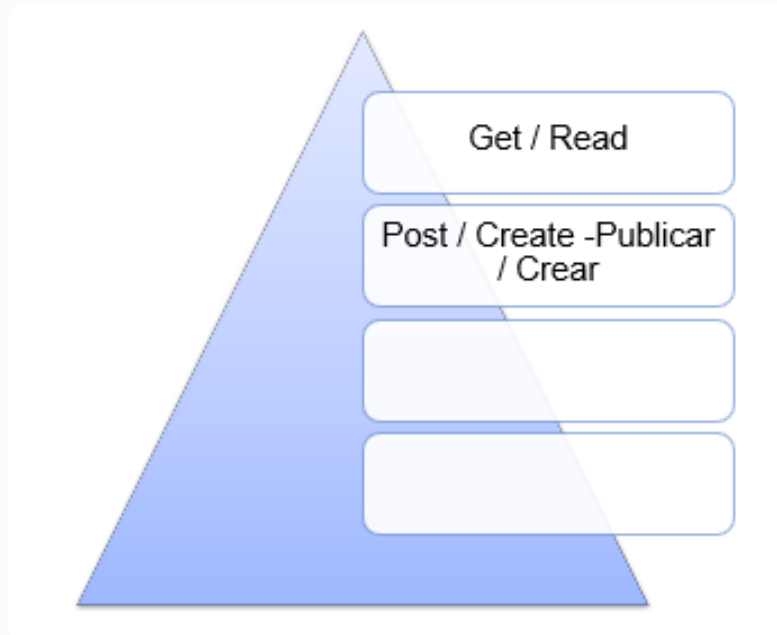
API Testing

- El método HTTP GET se utiliza para ****leer**** (o recuperar) una representación de un recurso. En el camino "feliz" (o sin error), GET devuelve una representación en XML o JSON y un código de respuesta HTTP de 200 (OK). En caso de error, suele devolver un 404 (NOT FOUND) o 400 (BAD REQUEST).



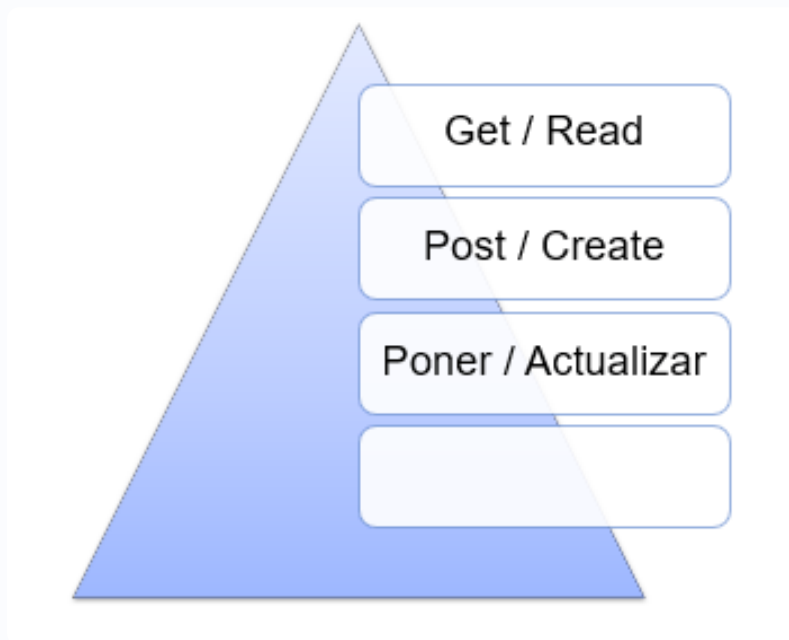
API Testing

- El verbo POST se utiliza sobre todo para **crear** nuevos recursos. En particular, se utiliza para crear recursos subordinados. En otras palabras, cuando se crea un nuevo recurso, POST al padre y el servicio se encarga de asociar el nuevo recurso con el padre, asignar un ID (URI del nuevo recurso), etc.
- Si la creación tiene éxito, devuelve el estado HTTP 201, devolviendo una cabecera Location con un enlace al recurso recién creado con el estado HTTP 201.



API Testing

- PUT se utiliza más a menudo para **actualizar** capacidades, PUT-ing a un recurso conocido URI con el cuerpo de la solicitud que contiene la representación recién actualizada del recurso original. Si la actualización se realiza correctamente, un PUT devuelve 200 (o 204 si no devuelve ningún contenido en el cuerpo).



API Testing

- Si la eliminación se realiza correctamente, devuelve el estado HTTP 200 (OK) junto con un cuerpo de respuesta, que puede ser la representación del elemento eliminado.
- Así que si quieres eliminar un usuario en tu aplicación (o el usuario quiere eliminar su propia cuenta), la aplicación hará una llamada a la API con el método /DELETE y el usuario será eliminado de la base de datos.

