



UNIVERSIDAD
CATÓLICA
BOLIVIANA

Tercer Parcial: Pruebas de API y Pruebas E2E en Todo.ly

Oscar Campohermoso Berdeja

Universidad Católica Boliviana

SIS-312: Gestión de Calidad de Sistemas

Profesor: Lic. Cecilia Alvarado Monrroy

10 de noviembre de 2024

Índice general

1. API Testing	2
1.1. Resumen de Ejecución de Pruebas	2
1.2. Descripción de las Pruebas	3
1.2.1. GET /Filters/Id	3
1.2.2. GET /Filters/Id/Items	4
1.2.3. GET /Filters/Id/DoneItems	5
1.2.4. GET /Items	6
1.3. Conclusión	7
2. Pruebas E2E	8
2.1. Casos de Prueba E2E	8
2.1.1. Creación de una Tarea	8
2.1.2. Completado de una Tarea	8

Capítulo 1

API Testing

Este capítulo describe el proceso de pruebas de API realizadas en el servicio de Todo.ly. Estas pruebas se llevaron a cabo para verificar la correcta funcionalidad, rendimiento y estructura de las respuestas de los endpoints seleccionados.

1.1. Resumen de Ejecución de Pruebas

Las pruebas se ejecutaron utilizando la herramienta Newman, que permite correr colecciones de Postman desde la línea de comandos. El siguiente es un resumen de la ejecución:

```
> newman run API-testing-OscarCampohermoso.postman_collection.json -e
QAenv-OscarCampohermoso.postman_environment.json --color off
```

```
API-testing-OscarCampohermoso
```

```
-> Crear Usuario
  | 'Generated Email:', 'taylor_johnson@example.com'
  | 'Generated Full Name:', 'Taylor Johnson'
  POST https://todo.ly/api/user.json [200 OK, 705B, 1037ms]

-> POST /Projects
  POST https://todo.ly/api/projects.json [200 OK, 721B, 229ms]

-> POST /Items 1rs item
  POST https://todo.ly/api/items.json [200 OK, 851B, 232ms]

-> POST /Items 2nd item
  POST https://todo.ly/api/items.json [200 OK, 851B, 247ms]

-> POST /Items 3rd item
  POST https://todo.ly/api/items.json [200 OK, 851B, 249ms]

-> PUT /Items/Id
  PUT https://todo.ly/api/items/11600340.json [200 OK, 870B, 232ms]
```

```

-> GET /Filters
GET https://todo.ly/api/filters.json [200 OK, 637B, 227ms]

-> * GET /Filters/Id
GET https://todo.ly/api/filters/-1.json [200 OK, 390B, 228ms]
- El código de estado es 200
- El tiempo de respuesta es menor a 2000ms
- El contenido del filtro es una cadena de texto
- El tipo de ítem es 4 (Filtro)
- El ID del filtro es un número negativo
- El número de ítems es un número entero

-> * GET /Filters/Id/Items
GET https://todo.ly/api/filters/-2/items.json [200 OK, 1.39kB, 234ms]
- El código de estado es 200
- El tiempo de respuesta es menor a 2000ms
- La respuesta es una lista de ítems
- Verificar que cada ítem tenga el OwnerId igual a {{user_id}}
- Verificar que cada ítem tenga un ID válido

-> * GET /Filters/Id/DoneItems
GET https://todo.ly/api/filters/-2/doneitems.json [200 OK, 872B, 228ms]
- El código de estado es 200
- El tiempo de respuesta es menor a 2000ms
- La respuesta es una lista de ítems completados
- Verificar que cada ítem tenga el OwnerId igual a {{user_id}}
- Verificar que cada ítem tenga un ID válido
- Verificar ítems completados

-> * GET /Items
GET https://todo.ly/api/items.json [200 OK, 1.95kB, 226ms]
- El código de estado es 200
- El tiempo de respuesta es menor a 2000ms
- La respuesta es una lista de ítems
- Verificar que cada ítem tenga el OwnerId igual a {{user_id}}
- Cada ítem tiene un ID y contenido válidos

```

1.2. Descripción de las Pruebas

Las pruebas se diseñaron para los siguientes endpoints, y como se observa en el resumen de la ejecución de pruebas, se llevaron a cabo con éxito. Cada prueba fue planificada para verificar tanto la estructura de las respuestas como la consistencia y validez de los datos retornados por la API.

1.2.1. GET /Filters/Id

Este endpoint devuelve los detalles de un filtro específico y se validaron los siguientes aspectos:

- Código de estado 200 ✓
- Tiempo de respuesta menor a 2000 ms ✓
- El contenido del filtro es una cadena de texto ✓
- El tipo de ítem es 4 (Filtro) ✓
- El ID del filtro es un número negativo ✓
- El número de ítems es un número entero ✓

Este endpoint permite al usuario obtener detalles sobre filtros específicos, como los que muestran las tareas de hoy o próximas. Verificar la integridad de estos datos es crucial para la correcta visualización de la información en la interfaz.

```

1 pm.test("El código de estado es 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test("El tiempo de respuesta es menor a 2000ms", function () {
6   pm.expect(pm.response.responseTime).to.be.below(2000);
7 });
8
9 pm.test("El contenido del filtro es una cadena de texto", function () {
10   var jsonData = pm.response.json();
11   pm.expect(jsonData.Content).to.be.a('string');
12 });
13
14 pm.test("El tipo de ítem es 4 (Filtro)", function () {
15   var jsonData = pm.response.json();
16   pm.expect(jsonData.ItemType).to.eqls(4);
17 });
18
19 pm.test("El ID del filtro es un número negativo", function () {
20   var jsonData = pm.response.json();
21   pm.expect(jsonData.Id).to.be.below(0);
22 });
23
24 pm.test("El número de ítems es un número entero", function () {
25   var jsonData = pm.response.json();
26   pm.expect(jsonData.ItemsCount).to.be.a('number');
27 });

```

1.2.2. GET /Filters/Id/Items

Este endpoint devuelve la lista de ítems asociados a un filtro específico. Las validaciones incluyen:

- Código de estado 200 ✓
- Tiempo de respuesta menor a 2000 ms ✓
- La respuesta es una lista de ítems ✓
- Cada ítem tiene un OwnerId igual a {{user_id}} ✓

- Cada ítem tiene un ID válido ✓

Este endpoint es útil para listar las tareas filtradas por un criterio específico, como las tareas que deben realizarse hoy. Las pruebas aseguran que cada ítem en la lista esté correctamente asociado al usuario autenticado.

```

1 pm.test("El código de estado es 200", function () {
2     pm.response.to.have.status(200);
3 });
4
5 pm.test("El tiempo de respuesta es menor a 2000ms", function () {
6     pm.expect(pm.response.responseTime).to.be.below(2000);
7 });
8
9 pm.test("La respuesta es una lista de ítems", function () {
10     var jsonData = pm.response.json();
11     pm.expect(jsonData).to.be.an('array');
12 });
13
14 pm.test("Verificar que cada ítem tenga el OwnerId igual a {{user_id}}", function () {
15     var jsonData = pm.response.json();
16     var userId = pm.variables.get("user_id");
17     jsonData.forEach(function(item) {
18         pm.expect(item.OwnerId).to.eql(parseInt(userId));
19     });
20 });
21
22 pm.test("Verificar que cada ítem tenga un ID válido", function () {
23     var jsonData = pm.response.json();
24     jsonData.forEach(function(item) {
25         pm.expect(item.Id).to.be.a('number').and.to.be.above(0);
26     });
27 });

```

1.2.3. GET /Filters/Id/DoneItems

Este endpoint devuelve la lista de ítems completados asociados a un filtro específico. Las validaciones realizadas fueron:

- Código de estado 200 ✓
- Tiempo de respuesta menor a 2000 ms ✓
- La respuesta es una lista de ítems completados ✓
- Cada ítem tiene un OwnerId igual a {{user_id}} ✓
- Cada ítem tiene un ID válido ✓
- Verificar que los ítems estén marcados como completados ✓

Este endpoint permite al usuario ver las tareas que ha completado. Las pruebas aseguran que todos los ítems devueltos estén efectivamente marcados como completados y asociados al usuario correcto.

```

1 pm.test("El código de estado es 200", function () {
2     pm.response.to.have.status(200);
3 });
4
5 pm.test("El tiempo de respuesta es menor a 2000ms", function () {
6     pm.expect(pm.response.responseTime).to.be.below(2000);
7 });
8
9 pm.test("La respuesta es una lista de ítems completados", function () {
10     var jsonData = pm.response.json();
11     pm.expect(jsonData).to.be.an('array');
12 });
13
14 pm.test("Verificar que cada ítem tenga el OwnerId igual a {{user_id}}", function () {
15     var jsonData = pm.response.json();
16     var userId = pm.variables.get("user_id");
17     jsonData.forEach(function(item) {
18         pm.expect(item.OwnerId).to.eqls(parseInt(userId));
19     });
20 });
21
22 pm.test("Verificar que cada ítem tenga un ID válido", function () {
23     var jsonData = pm.response.json();
24     jsonData.forEach(function(item) {
25         pm.expect(item.Id).to.be.a('number').and.to.be.above(0);
26     });
27 });
28
29 pm.test("Verificar ítems completados", function () {
30     var jsonData = pm.response.json();
31     jsonData.forEach(function(item) {
32         pm.expect(item.Checked).to.eqls(true);
33     });
34 });

```

1.2.4. GET /Items

Este endpoint devuelve la lista de todos los ítems del usuario autenticado. Las pruebas realizadas fueron:

- Código de estado 200 ✓
- Tiempo de respuesta menor a 2000 ms ✓
- La respuesta es una lista de ítems ✓
- Cada ítem tiene un OwnerId igual a {{user_id}} ✓
- Cada ítem tiene un ID y contenido válidos ✓

Este endpoint es fundamental para mostrar todas las tareas de un usuario y comprobar que los datos devueltos sean correctos y completos.

```

1 pm.test("El código de estado es 200", function () {
2     pm.response.to.have.status(200);

```

```

3 });
4
5 pm.test("El tiempo de respuesta es menor a 2000ms", function () {
6     pm.expect(pm.response.responseTime).to.be.below(2000);
7 });
8
9 pm.test("La respuesta es una lista de ítems", function () {
10     var jsonData = pm.response.json();
11     pm.expect(jsonData).to.be.an('array');
12 });
13
14 pm.test("Verificar que cada ítem tenga el OwnerId igual a {{user_id}}", function () {
15     var jsonData = pm.response.json();
16     var userId = pm.variables.get("user_id");
17     jsonData.forEach(function(item) {
18         pm.expect(item.OwnerId).to.eqls(parseInt(userId));
19     });
20 });
21
22 pm.test("Cada ítem tiene un ID y contenido válidos", function () {
23     var jsonData = pm.response.json();
24     jsonData.forEach(function(item) {
25         pm.expect(item.Id).to.be.a('number').and.to.be.above(0);
26         pm.expect(item.Content).to.be.a('string').and.to.not.be.empty;
27     });
28 });

```

1.3. Conclusión

Las pruebas realizadas sobre los endpoints de la API de Todo.ly han demostrado que el sistema responde correctamente bajo las condiciones esperadas, manteniendo un tiempo de respuesta adecuado y proporcionando datos precisos. Las validaciones realizadas aseguran que la integridad de los datos y la experiencia del usuario sean consistentes y fiables.

Capítulo 2

Pruebas E2E

Este capítulo aborda las pruebas end-to-end (E2E) diseñadas para verificar flujos completos de usuario en la aplicación Todo.ly. Estas pruebas aseguran que las diferentes funcionalidades trabajen correctamente en conjunto.

2.1. Casos de Prueba E2E

2.1.1. Creación de una Tarea

Prueba diseñada para confirmar que un usuario puede crear una nueva tarea y que esta se refleje en la lista de tareas.

2.1.2. Completado de una Tarea

Prueba que valida que un usuario pueda marcar una tarea como completada y que su estado se actualice correctamente en la interfaz.