Informe sobre los casos de prueba utilizados en Java Etherpad Lite Client

Sergio Fernández Fraga

Clasificación y análisis de los métodos

En esta parte del informe, ayudándonos de la teoría de Validación de Software, clasificaremos todos los casos de prueba que han sido usados en el proyecto de Java Etherpad Lite Client. Analizaremos ambas clases existentes en src/test/java.

La lista de los casos de prueba es la siguiente:

1. validate_token

En este caso de prueba se busca validar que el api key del cliente es correcto.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

2. create_and_delete_group

En este caso de prueba se testea la creacion y eliminacion de grupos.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

3. create_group_if_not_exists_for_and_list_all_groups

En este caso de prueba se testea varias funcionalidades adicionales de los grupos. Creación de grupo con nombre y listado de grupos.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?

D	N	P	F	No

4. create_group_pads_and_list_them

En este caso de prueba se testean los pads. Creación y listado.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

5. create_author

En este caso de prueba se testean autores.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

6. create_author_with_author_mapper

En este caso de prueba se testea varias funcionalidades adicionales de los usuarios. Creación y obtención de información del usuario.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

7. create_pads_and_list_them

En este caso de prueba se testean pads. Se crean dos pads y se comprueba en una lista adicional que realmente se han creado dos.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

8. create_and_delete_session

En este caso de prueba se testea las sesiones. Creación, listado de las sesiones de un grupo e información de autores en esas sesiones.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

9. create_pad_move_and_copy

En este caso de prueba se testean los pads. Creación de un pad, hacer una copia o un traslado del pad.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

10. create_pad_set_and_get_content

Todo lo relacionado a un pad. Creación de texto, obtener información, añadir texto, etc.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

11. create_pad_and_chat_about_it

Diversas operaciones sobre un pad son testeadas en este caso de prueba.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

12. domain_with_trailing_slash_when_construction_an_api_path

Se comprueba que añadiendo, intencionadamente, un slash al final del recurso solicitado funciona de manera correcta.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

13. domain_without_trailing_slash_when_construction_an_api_path

Se comprueba que al solicitar un recurso sin un slash al final de este, la solicitud es correcta

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

14. query_string_from_map

En este caso de prueba se testea que ante un petición especificada, se obtiene el recurso esperado.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

15. url_encoded_query_string_from_map

Es un caso similar al anterior.

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	N	P	F	No

16. api_url_need_to_be_absolute

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?

D	N	N	F	No

17. handle_valid_response_from_server

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	B	P	F	Si

18. handle_invalid_parameter_error_from_server

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	B	N	F	Si

19. handle_internal_error_from_server

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	B	N	F	Si

20. handle_no_such_function_error_from_server

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	B	N	F	Si

21. handle_invalid_key_error_from_server

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	B	P	F	Si

22. unparsable_response_from_the_server

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	B	N	F	Si

23. unexpected_response_from_the_server

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	B	N	F	Si

24. valid_response_with_null_data

¿Dinámicas, estáticas o simbólicas?	¿Caja blanca o negra?	¿Positiva o negativa?	¿Funcional o no funcional?	¿Estructural?
D	B	N	F	Si

Posibles carencias encontradas

En primer lugar, no existe ningún caso de prueba estático. Existen varios tipos de testeo estático, entre ellos estarían, análisis, inspección y “walkthroughts” de código. Normalmente se suele utilizar para detectar fallos estructurales que pueden llevar a defectos.

En segundo lugar, no existen pruebas no funcionales, aquellas encargadas de testear el rendimiento de nuestro sistema. Se encargan de revisar el rendimiento, usabilidad, confiabilidad de nuestra aplicación ya que afectan al usuario final. De igual forma que los casos de prueba analizados son pruebas funcionales, debería existir algún caso no funcional ya que ayudaría a reducir riesgos de producción y costes asociados a aspectos no funcionales del producto final.

Finalmente, en algunos casos de prueba, por ejemplo, create_pad_set_and_get_content, considero que se están probando demasiadas funcionalidades de nuestra aplicación. Esto hace que el caso de prueba sea díficil de crear, leer y mantener.

Reflexión sobre la cobertura del proyecto

A pesar de que disponemos una cobertura alta de pruebas, esto no implica que nuestro proyecto sea inmune a fallos. Lo único que implica es que nuestras pruebas han recorrido

una gran parte del conjunto de líneas de nuestro código, es en el buen planteamiento de cada caso de prueba dónde deberíamos enfocarnos para lograr un buen testeo de nuestra aplicación. Un sólo caso de prueba siempre debería centrarse en un solo aspecto de nuestra aplicación. Conlleva más tiempo y mantenimiento hacer un caso de prueba grande que puede que sea satisfactorio pero tiene más inconvenientes que ventajas.