

Test Document

Objecte de la prova

En aquest test unitari estem testejant la classe document i els seus mètodes públics. El principal objectiu d'aquesta classe es mantenir les dades referents a cada document donat d'alta en el sistema i, a més, implementar els algorismes que seran usats per als casos d'ús de "llistar documents per Query i "llistar k similars".

Altres elements integrats a la prova

No hi ha cap element integrat.

Drivers

No s'usa cap driver en aquest test.

Stubs

La classe Document fa ús de la classe InternalDocument i, per tant, ens caldrà un stub d'aquesta per a ferne el test unitari. Aquest stub ha estat implementat de la manera més senzilla possible per a obtenir els resultats esperats de retorn del nostre test.

Fitxers de dades necessaris

No es requereix cap fitxer addicional.

Valors estudiats

Per a aquest tests, hem de comprovar diversos casos i funcions. Primer de tot, els getters i els setters, així com que aquests llencin les excepcions correctes quan calgui.

- Getters

El primer test a realitzar es crear un document amb la primera de les creadores de documents, assignat-li un autor, un títol, un contingut, un llenguatge i un format. Per a provar els getters, crearem un nou document assignat-li autor "Paula", títol "Títol del document", contingut "Tres tristes, tristes tigres comen trigo en un trigal", idioma "en" i format "txt", que son formats i idiomes vàlids que ens permeten crear el document sense excepcions i, per tant, validar-ne a continuació els getters. Repetim el mateix procediment per a la constructora de Document que no assigna cap format, i comprovem la correctesa dels resultats obtinguts pels getters.

- Setters

Per a comprovar els setters, només cal crear un document amb paràmetres vàlids, i invocar les dues funcions a provar, `setLanguage()` i `setContent()`, amb els nous valors a assignar. Un cop assignats els nous valors, utilitzem els getters per a comprovar que aquests s'han actualitzat correctament.

- Comprovació d'excepcions

Els següents tests corresponen a comprovar que donats valor invàlids d'idioma o de format, la creadora de document llença l'excepció corresponent. Primer comprovem que al crear un document amb idioma "abcd", efectivament es llença una excepció. A continuació, creem un document amb idioma vàlid i comprovem que el setter de l'atribut `language` també llença excepció si intentem canviar l'idioma del document a "abcd". Repetim exactament els mateixos tests per a l'atribut del format del document i la seva corresponent excepció.

A part d'això, necessitem comprovar el funcionament dels algorismes que ens permetran operar amb Document desde el sistema, concretament funcions públiques de comparacions entre documents i de rellevància de querys. Per a fer comparacions entre documents, ens cal comprovar les dues versions, una per a cada algoritme d'assignació de pesos del sistema.

- Test de `queryRelevance()`

Creem un document nou amb autor "Paula", títol "t1", contingut "Tres tristes, tristes tigres comen trigo en un trigal", idioma i format vàlid. Comprovem que l'algorisme retorna el valor esperat per a diverses querys. Per exemple, "Mi mama me mima" retornarà 0, ja que cap de les paraules claus coincideix. En canvi, "tigres trigo", retornarà un valor decimal major que 0 corresponent a aplicar l'algorisme de `tf-idf` a les paraules de tigres i trigo. També cal comprovar que si afegim stopwords a la query, el resultat d'aquesta no varia.

- Test de `compare_tf_idf()`

Per a aquest test, crearem diversos documents amb valors vàlids d'idioma i format i continguts dels que fàcilment podem avaluar-ne la similaritat. Per exemple, podem crear documents amb continguts "Tres tristes, tristes tigres comen trigo en un trigal", "Mi mama me mima", "Tres tristes palabras" etc. Per a fer aquest tests, comprovem que l'algorisme ens retorna el resultat numèric esperat en executar el mètode públic `compare_tf_idf` per a cadascun dels documents creats. També comprovem aquí que si dos documents tenen diferent idioma, l'algorisme retornarà similaritat igual a 0.

- Test de compare_tf_boolean

La metodologia de testing i casos comprovats és la mateixa que en el test de l'algorisme anterior, exceptuant que els valors numèrics resultants seran diferents.

Efectes estudiats

No aplica en aquest test, que no és de la capa de presentació.

Operativa

Per executar aquest test, cal introduir per consola l'ordre d'execució de java usant les llibreries de junit i hamcrest. Si es realitza des del directori arrel del projecte, és:

```
# java -cp ./EXE/Document:/lib/junit-4.12.jar:/lib/hamcrest-core-1.3.jar  
org.junit.runner.JUnitCore test.domain.documents.TestDocument
```