

Gestió de fitxers XML amb DOM, SAX i XStream

Aniol Moreno Batlle

DAM2

XSTREAM

Avantatges:

- En els fòrums es diu que és més ràpid que el DOM.
- És una bona opció per fitxers que pesin molt amb molta quantitat.
- Si s'utilitza objectes en l'aplicació és una bona pràctica, ja que es fa amb objectes i el codi et pot quedar molt més endreçat.
- Permet editar el nom de les etiquetes, o crear n'he de noves.

Desavantatges:

- Molt complexa d'utilitzar, si fa temps que no toques objectes en Java es fa molt complicat.
- Es necessiten moltes llibreries.
- Els permisos, no funciona si no li dones els permisos adequats en el fitxer que vols llegir.
- La documentació és pobre, costa trobar les coses i li falten exemples d'ús.
- Si tens el primer element que no es diu List ho has d'especificar, si no peta el codi.
- S'ha de tocar el pom.

DOM

Avantatges:

- És fàcil d'utilitzar per llegir.
- Bastant intuïtiu.
- És d'accés aleatori.
- Documentació més ampla que el XStream, ja que és de Java en si.

Gestió de fitxers XML amb DOM, SAX i XStream

Desavantatges:

- No serveix per fitxers grans, està pensat per fitxers petits.
- Té un rendiment lent en comparació a XStream i SAX.
- Per escriure no és tan fàcil com llegir, es necessita més codi i una mica més complexa.
- Quan afegeixo text en un arxiu ja existent, el text ja existent em posa amb un enter o dos perquè sí, tot i haver-ho indentat prèviament.
- Si no es normalitza el document no funciona el codi.

SAX

Avantatges:

- Té el rendiment més ràpid del tres.
- Documentació més ampla que el XStream, ja que és de Java en si.

Desavantatges:

- No és aleatori, així que no pots accedir a una part de l'XML en concret sense haver de passar per la resta.
- Només és de lectura, no d'escriure.
- Es necessita un Handler i això complica les coses, la part del Handler costa una mica d'entendre.