

Capa de datos

PROP Grupo 33
Víctor Muñoz, Alex Moa, Artur Farriols, Mélanie Scaténa
Cuatrimestre otoño
Curso 21-22

Introducción

Este documento incluye una descripción sucinta de las clases y *enumerations* pertenecientes a la capa de datos (persistencia). Al tratarse de un trabajo iterativo e incremental, la información aquí proporcionada está sujeta al cambio en función de las modificaciones aconsejadas por el tutor o aquellas decididas por los integrantes del grupo.

Controllers

CrtlData

Descripción: representa el controlador de la capa de datos por lo que se encarga de comunicarse con la capa de dominio. Es por ello que contiene todas las funciones necesarias para garantizar la persistencia de los datos.

Variables:

- DataUtils utils

Funciones y métodos:

Lectura:

- `Map<String, Item> loadItems(String dataset)`
Para un dataset existente, devuelve un diccionario con clave ítem id y valor el propio ítem. En caso de no existir el dataset o hubiese una excepción devolverá un diccionario nulo.
- `Map<String, User> loadUsers(String dataset, String filename, Map<String,Item> items):`
Para un dataset y filename existentes, junto a un diccionario de ítems, retorna un *map* con clave user id y valor el propio usuario. En caso de no existir el dataset o producirse una excepción, devolverá un diccionario nulo.
Cualquier método que cargue usuarios hace una llamada a este método.
- `Map<String,User> loadUsers(String dataset, Map<String,Item> items):`
Para un dataset existente, junto a un diccionario de ítems, retorna un *map* de los usuarios con clave user id y valor el propio usuario. En caso de no existir el dataset o producirse una excepción devolverá un diccionario nulo.

- `Map<String, User> loadKnownUsers(String dataset, Map<String, Item> items)`
Para un dataset existente, junto a un diccionario de ítems, retorna un *map* de los usuarios *known* con clave user id y valor el propio usuario. En caso de no existir el dataset o producirse una excepción devolverá un diccionario nulo.
- `public Map<String, User> loadUnknownUsers(String dataset, Map<String, Item> items)`
Para un dataset existente, junto a un diccionario de ítems, retorna un *map* de los usuarios *unknown* con clave user id y valor el propio usuario. En caso de no existir el dataset o producirse una excepción devolverá un diccionario nulo.
- `Map<String, Recommendation> loadRecommendations(String dataset, Map<String, User> users)`
Para un dataset existente, junto a un diccionario de usuarios, retorna un *map* de las recomendaciones hechas con clave recommendation id y valor la propia recomendación. En caso de no existir el dataset o producirse una excepción devolverá un diccionario nulo.

Escritura:

Todas las escrituras, a excepción de las recomendaciones, se realizan en el subdirectorio **temp** aún sí el dataset dado es el propio **temp**. Dicho de otro modo, los ficheros en la raíz del dataset se usan a modo de plantilla.

- `boolean writeItems(String dataset, Item[] items):`
Para un dataset existente, junto a un *array* de ítems, retorna **true** en caso que se hayan podido escribir correctamente y **false** en caso contrario.
- `boolean writeNewItem(String dataset, Item item):`
Para un dataset existente, junto a un ítem, retorna **true** en caso que se haya podido escribir correctamente y **false** en caso contrario.
- `boolean writeUsers(String dataset, User[] users):`
Para un dataset existente, junto a un *array* de usuarios, retorna **true** en caso que se hayan podido escribir correctamente y **false** en caso contrario.
- `boolean writeKnownUsers(String dataset, User[] users):`
Para un dataset existente, junto a un *array* de usuarios *known*, retorna **true** en caso que se hayan podido escribir correctamente y **false** en caso contrario.

- `boolean writeUnknownUsers (String dataset, User[] users):`
Para un dataset existente, junto a un *array* de usuarios *unknown*, retorna **true** en caso que se hayan podido escribir correctamente y **false** en caso contrario.
- `boolean writeRecommendations (String dataset, Recommendation[] recommendations):`
Para un dataset existente, junto a un *array* de recomendaciones, retorna **true** en caso que se hayan podido escribir correctamente y **false** en caso contrario.
- `boolean writeNewRecommendations (String dataset, Recommendation recommendation):`
Para un dataset existente, junto a una recomendación, retorna **true** en caso que se haya podido escribir correctamente y **false** en caso contrario.

Otras:

- `String[] getDatasets():`
Devuelve el listado de datasets disponibles en el sistema y que contengan los ficheros:
 - items.csv
 - ratings.db.csv
 - ratings.test.known.csv
 - ratings.test.unknown.csv
- `boolean existTemp (String dataset):`
Devuelve **true** en caso que dicho dataset contenga el subdirectorio **temp** con los 4 ficheros necesarios y **false** en caso contrario.
- `String getPath (String dataset):`
Devuelve el path hacía el dataset.

Utils

CSVReader

Descripción: clase que, a partir de un *path* realizará la lectura de los ficheros csv, es por ello que no utilizará ninguna clase del modelo.

Variables:

- String DELIMITER (valor que separa los atributos de un fichero csv)

Funciones y métodos:

- `String[] readFirstLine(String path):`
Dado un path hacía un fichero csv, devuelve los atributos de la cabecera de un fichero csv.
- `String[] readLine(String line, String delimiter):`
Devuelve los atributos de línea separados por el delimitador.
- `String[][] readFile(String path):`
Dado un path hacía un fichero csv, devuelve el conjunto de atributos separados por el delimitador de cada línea del fichero.

DataUtils

Descripción: clase encargada de transformar los modelos del sistema en un formato válido para la persistencia de datos en un fichero csv y viceversa.

Funciones y métodos:

- `Map<String, Item> getItems(String path):`
Dado un path hacía un fichero csv, retorna un diccionario con clave el item id y valor el propio ítem. En caso de ocurrir algún error devolverá un mapa nulo.
- `Map<String, User> getUsers(String path, Map<String, Item> items):`

Dado un path hacía un fichero csv y un mapa de ítems, retorna un diccionario con clave el user id y valor el propio usuario. En caso de ocurrir algún error devolverá un mapa nulo.

- `Map<String, Recommendation> getRecommendations(String path, Map<String, User> users):`

Dado un path hacía un fichero csv y un mapa de usuarios, retorna un diccionario con clave el recommendation id y valor la propia recomendación. En caso de ocurrir algún error devolverá un mapa nulo.

- `void writeTempItems(String path, Item[] items):`

Dado un path hacía un fichero, escribe el listado de ítems. En caso de producirse alguna excepción, ésta se propaga.

- `void writeTempItem(String path, Item item):`

Dado un path hacía un fichero, escribe el ítem. En caso de producirse alguna excepción, ésta se propaga.

- `void writeUserHeader(FileWriter fileWriter):`

Escribe la cabecera del usuario usando el FileWriter dado.

- `void writeUserRates(FileWriter fileWriter, User user):`

Escribe al usuario usando el FileWriter dado.

- `void writeNewTempUser(String path, User user):`

Dado un path hacía un fichero, escribe el usuario. En caso de producirse alguna excepción, ésta se propaga.

- `void writeTempUsers(String path, User[] users):`

Dado un path hacía un fichero, escribe el listado de usuarios. En caso de producir alguna excepción, ésta se propaga.

- `void writeRecommendations(String path, Recommendation[] recommendations):`

Dado un path hacía un fichero, escribe el listado de recomendaciones. En caso de producirse alguna excepción, ésta se propaga.

- `void writeRecommendation(FileWriter fileWriter, Recommendation recommendation)`

Escribe la recomendación usando el FileWriter dado.

- `void writeNewRecommendation(String path, Recommendation recommendation)`
Dado un path hacía un fichero, escribe la recomendación. En caso de producirse alguna excepción, ésta se propaga.
- `boolean existDataset(String path)`
Dado un path, retorna **true** si este existe y contiene los 4 ficheros necesarios y **false** en caso contrario.
- `String[] getAttributes(String path)`
Dado un path hacía un fichero, retorna la cabecera de este.

Utils

Descripción: clase estática empleada para almacenar variables de entorno del sistema necesarias para el correcto funcionamiento de la capa de datos.

Variables:

- String PATH.
- String ITEMS.
- String USERS.
- String KNOWN_USERS.
- String UNKNOWN_USERS.
- String TEMP.
- String RECOMMENDATIONS.
- String LINE_BREAK.