

Visualization KORONA

Pre-processing

This project requires an additional external library in order to enable access and processing of JSON-objects:

- `javax.json-1.1.2`
-

ListToJS.java

Program description

Converts a text-file containing ID and names of authors into a JSON-object. In this way the information to visualize a select list is provided.

Required files: ▪ `selectauthor.txt`

Output: ▪ `selectauthor.js`

Methods

`ArrayList<String[]> retrieveList(String path)`

Retrieves the list stored in the file corresponding to path. Each entry in the retrieved list consists of an array with two entries; the first entry contains an author's ID, the second entry contains an author's name.

`String convertList(ArrayList<String[]> authorlist, String name)`

Converts the list `authorlist` into a JSON-object and assigns it to a variable `name`. The result is then stored in a `String`.

`void writeToNewFile(String path, String content)`

Creates a new file using the given path and stores the text given as `String content`.

Program Flow (main method)

- The input-files are processed
- The list stored in the input-file is retrieved (`retrieveList()`)
- The path and name of the output-file and the variable name of the JSON-object to be created are assigned
- The retrieved list is converted into a JSON-object (`convertList()`)
- The output-file is created (`writeToNewFile()`)

ClustersToJS.java

Program description

The root-directory given as input contains subdirectory each corresponding to an author. Each subdirectory contains text-files which correspond to clusters in a network. This program converts the given file-structure and content into JSON-objects and stores them in a JS-file.

Input:

- `authorkeys.txt`
- `selectedpredictions/`
- `suffix` (i.e., `s95`)

Output:

- `network_suffix.js`

Data structure

static String authorkeyspath Path to authors-keys-file

static String suffix Suffix for output-files and produced JSON-object-names

Author (String id, ArrayList<Cluster> clusters)

Represents an author and relates to one of the given subdirectories. Each author has an ID and the list clusters refers to the text-files stored in the corresponding subdirectory.

Cluster (ArrayList<Node> nodes, ArrayList<Link> links)

Represents a cluster and consists of a list of nodes and a list of links which contains the relations between the nodes.

Node (String id, String name, int group)

Refers to an entity in a cluster, which corresponds to either an author or a conference edition. The IDs of authors start with the character "A", the first character of conference editions' IDs is "C". The attribute group specifies the kind of entity further; group 1 refers to the author who relates to the corresponding subdirectory, group 2 refers to other authors, and group 3 refers to conference editions. The distinction of different groups is required for the visualization of the network.

Link (String source, String target, double weight)

Represents the association of two entities with source always referring to an author-entity and target always referring to a conference edition-entity. The attribute weight refers to the similarity value of the author corresponding to source and the conference edition corresponding to target.

Methods

ArrayList<Cluster> readSubDir(String dirpath, String subdir)

Converts the cluster-files related to the author corresponding to subdir in a list of type Cluster.

- The files in the subdirectory are retrieved
- Each file is accessed and the corresponding cluster is retrieved:
 - The list of links is retrieved by accessing and processing the file
 - The list of nodes is retrieved by converting the information stored in the list of links
 - The cluster corresponding to this file is created and added to the list of clusters

void writeToNewFile(String path, String content)

Creates a new file using the given path and stores the text given as String content.

Program Flow (main method)

- Input-files and the suffix given as input are processed
- A list of subdirectories is retrieved
- Each subdirectory is converted into a JSON-object (readSubDir())
- The output-file containing all created JSON-objects is created (writeToNewFile())