

UFSCAR

---

## Zenodotus - Quick Guide

---

## Contents

<b>1</b>	<b>Introduction and Download</b>	<b>2</b>
<b>2</b>	<b>Methods</b>	<b>2</b>
2.1	List Getters . . . . .	2
2.1.1	Get Categories List . . . . .	2
2.1.2	Get Relations List . . . . .	3
2.2	Getters Methods . . . . .	4
2.2.1	Get Category . . . . .	4
2.2.2	Get Relation . . . . .	5
2.2.3	Get Category Instance . . . . .	7
2.3	Searchers Methods . . . . .	9
2.3.1	Search Category . . . . .	9
2.3.2	Search Relation . . . . .	10
2.3.3	Search Category Instance . . . . .	12

# 1 Introduction and Download

Zenodotus is a project in development and can be downloaded at:

<https://github.com/wesleywilly/Zenodotus/blob/master/Zenodotus/zenodotus.jar>

## 2 Methods





### 2.1 List Getters

#### 2.1.1 Get Categories List

Returns a JSONArray with all NELL's categories of KB beliefs.

**JSON structure**

Figure 1: Categories List Structure

Key	Type	Value
▼  Object	Object	↕
▼  categories_list	Array	↕
▼  Item[0]	Object	↕
A category_name	String	↕ everypromotedthing
A link	String	↕ http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=everypromo.
▶  Item[1]	Object	↕

**Java API**

```

1 package zenodotusexampleofuse;
2
3 import WebCrawlers.Zenodotus;
4 import org.json.simple.JSONArray;
5
6 public class CategoriesList {
7
8     public static void main(String [] args) {
9         JSONArray categoriesList = Zenodotus.getCategoriesList();
10    }
11 }
```

## Command-line

Calling Get Categories List and showing in command-interface

```
1 java -jar zenodotus.jar -gcl
```

Calling Get Categories List and saving results in a file







```
1 java -jar zenodotus.jar -gcl -s result.json
```

### 2.1.2 Get Relations List

Returns a JSONArray with all NELL's relations of KB beliefs.

## JSON structure

Figure 2: Relations List Structure

Key	Type	Value
▼  Object	Object	↕
▼  relations_list	Array	↕
▼  Item[0]	Object	↕
 link	String	↕ <a href="http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=relatedto">http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=relatedto</a>
 relation_name	String	↕ relatedto
▶  Item[1]	Object	↕

## Java API

```
1 package zenodotusexampleofuse;
2
3 import WebCrawlers.Zenodotus;
4 import org.json.simple.JSONArray;
5
6 public class CategoriesList {
7
8     public static void main(String[] args) {
9         JSONArray relationsList = Zenodotus.getRelationsList();
10    }
11 }
```

## Command-line

Calling Get Relations List and showing in command-interface

```
1 java -jar zenodotus.jar -grl
```

Calling Get Relations List and saving results in a file

```
1 java -jar zenodotus.jar -grl -s result.json
```

## 2.2 Getters Methods

### 2.2.1 Get Category

Returns a JSONObject represents an specific category with its instances.

#### Parameter:

- Predicate: Is the category name as it's known by NELL.

#### JSON structure

Figure 3: Category Structure







Key	Type	Value
▼  Object	Object	↕
📄 category_name	String	↕ movie
▶  category_instances	Array	↕
📄 category_prologue	String	↕ A release of a visual narrative composed of a series of moving i...
📄 link	String	↕ <a href="http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=movie">http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=movie</a>

Figure 4: Category Structure with Instances structure

Key	Type	Value
▼  Object	Object	↕
📄 category_name	String	↕ movie
▼  category_instances	Array	↕
▼  Item[0]	Object	↕
📄 instance_confidence	String	↕ (Seed) 100.0
📄 instance_name	String	↕ action_movies
📄 link	String	↕ <a href="http://rtw.ml.cmu.edu/rtw/kbbrowser/entity.php?id=movie:actio...">http://rtw.ml.cmu.edu/rtw/kbbrowser/entity.php?id=movie:actio...</a>
📄 instance_iteration	String	↕ 1018
📄 instance_data_learned	String	↕ 06-oct-2016
▶  Item[1]	Object	↕

## Java API

```
1 package zenodotusexampleofuse;
2
3 import WebCrawlers.Zenodotus;
4 import org.json.simple.JSONArray;
5 import org.json.simple.JSONObject;
6
7 public class CategoriesList {
8
9     public static void main(String[] args) {
10         JSONObject category = Zenodotus.getCategory("movie");
11     }
12 }
```

## Command-line

Calling *Get Category* and showing in command-interface

```
1 java -jar zenodotus.jar -gc movie
```

Calling *Get Category* and saving results in a file

```
1 java -jar zenodotus.jar -gc movie -s result.json
```

### 2.2.2 Get Relation

Returns a `JSONObject` represents an specific relation with its instances.

#### Parameter:

- Predicate: Is the relation name as it's known by NELL.

## JSON structure

Figure 5: Relation Structure












Key	Type	Value
▼  Object	Object ↕	
A relation_name	String ↕	actorstarredinmovie
A link	String ↕	<a href="http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=actorstarred...">http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=actorstarred...</a>
A relation_prologue	String ↕	to perform as an actor (in movie) ( <a href="http://dictionary.reference.co...">http://dictionary.reference.co...</a>
▼  relation_domian	Object ↕	
A category_name	String ↕	actor
A link	String ↕	<a href="http://rtw.ml.cmu.edu/rtw/kbbrowser/pred:actor">http://rtw.ml.cmu.edu/rtw/kbbrowser/pred:actor</a>
▼  relation_range	Object ↕	
A category_name	String ↕	movie
A link	String ↕	<a href="http://rtw.ml.cmu.edu/rtw/kbbrowser/pred:movie">http://rtw.ml.cmu.edu/rtw/kbbrowser/pred:movie</a>
▶  relation_instances	Array ↕	
▼  metadata	Object ↕	
▼  extractionPatterns	Array ↕	
A Item[0]	String ↕	arg1 action flick arg2
A Item[1]	String ↕	arg1 after the release of his arg2
A Item[2]	String ↕	arg1 agreed to star in arg2
A Item[3]	String ↕	arg1 and Anne Hathaway in arg2

Figure 6: Relation instances structure

▼  relation_instances	Array ↕	
▼  Item[0]	Object ↕	
A instance_confidence	String ↕	100.0
A instance_arg1	String ↕	al_pacino
A instance_arg2	String ↕	righteous_kill
A instance_iteration	String ↕	1042
A instance_data_learned	String ↕	05-mar-2017
▶  Item[1]	Object ↕	
▶  Item[2]	Object ↕	
▶  Item[3]	Object ↕	

## Java API

```
1 package zenodotusexampleofuse;
2
3 import WebCrawlers.Zenodotus;
4 import org.json.simple.JSONArray;
5 import org.json.simple.JSONObject;
6
7 public class CategoriesList {
8
9     public static void main(String[] args) {
10         JSONObject relation = Zenodotus.getRelation("
11         actorstarredinmovie");
12     }
```

## Command-line

Calling *Get Relation* and showing in command-interface

```
1 java -jar zenodotus.jar -gr actorstarredinmovie
```

Calling *Get Relation* and saving results in a file

```
1 java -jar zenodotus.jar -gr actorstarredinmovie -s result.json
```

### 2.2.3 Get Category Instance

Returns a `JSONObject` represents an specific instance of an specific category.

#### Parameter:

- Category Predicate: Is the category name as it's known by NELL.
- Instance Predicate: Is the category instance name as it's known by NELL.



## JSON structure

Figure 7: Category Instance structure








Key	Type	Value
▼  Object	Object	↕
A instance_name	String	↕ mel_gibson
▶  instance_literal_strings	Array	↕
A link	String	↕ <a href="http://rtw.ml.cmu.edu/rtw/kbbrowser/entity.php?id=actor:mel_gi...">http://rtw.ml.cmu.edu/rtw/kbbrowser/entity.php?id=actor:mel_gi...</a>
A instance_data_learned	String	↕ 05-mar-2017
A instance_iteration	String	↕ 1042
A instance_confidence	String	↕ 100.0
▼  instance_category	Object	↕
A category_name	String	↕ actor
▼  instance_category_cpl	Array	↕
A Item[0]	String	↕ movie stars _
A Item[1]	String	↕ movies starring _
A category_prologue	String	↕ An actor is a person who performs in movies, television, plays, o...
A link	String	↕ <a href="http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=actor">http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=actor</a>
▼  instance_category_seal	Array	↕
A Item[0]	String	↕ <a href="http://www.gmrnet.com/resultsActorRatings01.html">http://www.gmrnet.com/resultsActorRatings01.html</a>
A Item[1]	String	↕ <a href="http://www.cambridge1.net/Actors.html">http://www.cambridge1.net/Actors.html</a>
A Item[2]	String	↕ <a href="http://www.gmrnet.com/resultsActorRatings.html">http://www.gmrnet.com/resultsActorRatings.html</a>
A Item[3]	String	↕ <a href="http://content.usatoday.com/topics/index/People/Celebrities/Act...">http://content.usatoday.com/topics/index/People/Celebrities/Act...</a>
A Item[4]	String	↕ <a href="http://www.portaldecinema.com.br/lista_atores.htm">http://www.portaldecinema.com.br/lista_atores.htm</a>
A Item[5]	String	↕ <a href="http://www.chiff.com/pop-culture/hollywood.htm">http://www.chiff.com/pop-culture/hollywood.htm</a>
▼  instance_relations	Array	↕
▶  Item[0]	Object	↕

Figure 8: Literal String structure






▼  instance_literal_strings	Array	↕
▼  Item[0]	Object	↕
A literal_string_value	String	↕ Mel Gibson
A link	String	↕ <a href="http://www.google.com/search?q=%22Mel+Gibson%22">http://www.google.com/search?q=%22Mel+Gibson%22</a>
▶  Item[1]	Object	↕

Figure 9: Instance Relations structure

▼  instance_relations	Array	↕	
▼  Item[0]	Object	↕	
A link	String	↕	http://rtw.ml.cmu.edu/rtw/kbbrowser//pred:male
A relation_name	String	↕	male
▶  Item[1]	Object	↕	

## Java API

```

1 package zenodotusexampleofuse;
2
3 import WebCrawlers.Zenodotus;
4 import org.json.simple.JSONArray;
5 import org.json.simple.JSONObject;
6
7 public class CategoriesList {
8
9     public static void main(String[] args) {
10         JSONObject instance = Zenodotus.getCategoryInstance("actor", "
mel_gibson");
11     }
12 }

```

## Command-line

Calling *Get Category Instance* and showing in command-interface

```
1 java -jar zenodotus.jar -gci actor mel_gibson
```

Calling *Get Category Instance* and saving results in a file

```
1 java -jar zenodotus.jar -gci actor mel_gibson -s result.json
```

## 2.3 Searchers Methods

### 2.3.1 Search Category






Returns a JSONArray with found categories given a keyword. If keyword match with a known category name, it will return in the first position of list.

#### Parameter:

- Keyword: The method will return results based on this parameter.

## JSON structure

Figure 10: Search Result Structure

Key	Type	Value
▼  Object	Object ↕	
▼  categories_founded	Array ↕	
▼  Item[0]	Object ↕	
A category_name	String ↕	actor
▶  category_instances	Array ↕	
A category_prologue	String ↕	An actor is a person who performs in movies, television, plays, o...
A link	String ↕	<a href="http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=actor">http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=actor</a>
▶  Item[1]	Object ↕	

## Java API

```

1 package zenodotusexampleofuse;
2
3 import WebCrawlers.Zenodotus;
4 import org.json.simple.JSONArray;
5 import org.json.simple.JSONObject;
6
7 public class CategoriesList {
8
9     public static void main(String[] args) {
10         JSONArray categories = Zenodotus.searchCategory("movie");
11     }
12 }

```

## Command-line

Calling *Search Category* and showing in command-interface

```
1 java -jar zenodotus.jar -sc actor
```

Calling *Search Category* and saving results in a file

```
1 java -jar zenodotus.jar -sc actor -s result.json
```

### 2.3.2 Search Relation









Returns a `JSONArray` with found relations given a keyword. If keyword match with a known relation name, it will return in the first position of list.

**Parameter:**

- **Keyword:** The method will return results based on this parameter.

**JSON structure**

Figure 11: Search Result Structure

Key	Type	Value
▼  Object	Object ↕	
▼  relations_founded	Array ↕	
▼  Item[0]	Object ↕	
A relation_name	String ↕	moviessuchasmovies
A link	String ↕	http://rtw.ml.cmu.edu/rtw/kbbrowser/list.php?pred=moviessuch...
A relation_prologue	String ↕	Describes the movies types that are generalizations of specific...
▶  relation_domian	Object ↕	
▶  relation_range	Object ↕	
▶  metadata	Object ↕	
▶  relation_instances	Array ↕	
▶  Item[1]	Object ↕	

**Java API**

```

1 package zenodotusexampleofuse;
2
3 import WebCrawlers.Zenodotus;
4 import org.json.simple.JSONArray;
5 import org.json.simple.JSONObject;
6
7 public class CategoriesList {
8
9     public static void main(String[] args) {
10         JSONArray relations = Zenodotus.searchRelation("movie");
11     }
12 }

```

**Command-line**

Calling *Search Relation* and showing in command-interface

```
1 java -jar zenodotus.jar -sr movie
```

Calling *Search Relation* and saving results in a file

```
1 java -jar zenodotus.jar -sr movie -s result.json
```

### 2.3.3 Search Category Instance






Returns a JSONArray with found category instances given a keyword. If keyword match with a known relation name, it will return in the first position of list. This method is overloaded, is possible to search in two ways, with and without the category name.

#### Parameter:

- Predicate (optional): Is the category name as it's known by NELL.
- Keyword: The method will return results based on this parameter.

#### JSON structure

Figure 12: Search Result Structure

Key	Type	Value
▼  Object	Object ↕	
▼  instances_founded	Array ↕	
▼  Item[0]	Object ↕	
A instance_confidence	String ↕	100.0
A instance_name	String ↕	mel_gibson
A link	String ↕	<a href="http://rtw.ml.cmu.edu/rtw/kbbrowser/entity.php?id=personcana...">http://rtw.ml.cmu.edu/rtw/kbbrowser/entity.php?id=personcana...</a>
▼  instance_relations	Array ↕	
A instance_iteration	String ↕	1042
A instance_data_learned	String ↕	05-mar-2017
▶  Item[1]	Object ↕	

#### Java API

```

1 package zenodotusexampleofuse;
2
3 import WebCrawlers.Zenodotus;
4 import org.json.simple.JSONArray;
5 import org.json.simple.JSONObject;
6
7 public class CategoriesList {
8
9     public static void main(String[] args) {
10         JSONArray instances = Zenodotus.searchCategoryInstance("matrix");
11     }
12 }
```

```
12     JSONArray actorInstances = Zenodotus.searchCategoryInstance("actor"  
13     , "al pacino");  
14 }
```

## Command-line

Calling *Search Category Instance* without category name and showing in command-interface

```
1 java -jar zenodotus.jar -sci mel_gibson
```

Calling *Search Category Instance* without category name and saving results in a file

```
1 java -jar zenodotus.jar -sci mel_gibson -s result.json
```

Calling *Search Category Instance* with category name and showing in command-interface

```
1 java -jar zenodotus.jar -sci actor mel_gibson
```

Calling *Search Category Instance* with category name and saving results in a file

```
1 java -jar zenodotus.jar -sci actor mel_gibson -s result.json
```