

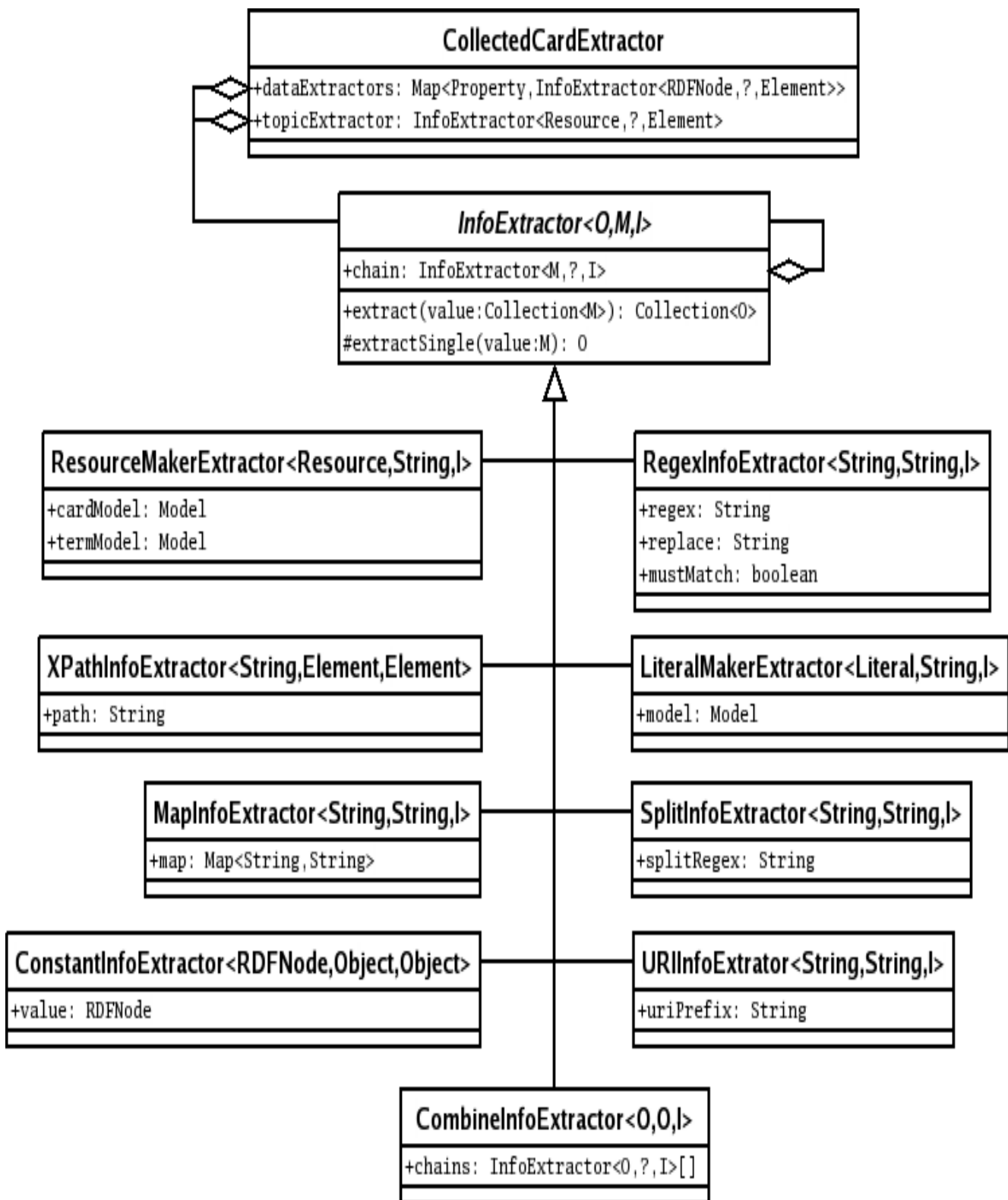
XML2RDF

Orava

Package

fi.helsinki.cs.orava.xml2rdf

XML2RDF package contains tools to perform conversion from XML file(s) into RDF. Conversion is done using the `CollectedCardExtractor` class. It uses multiple `InfoExtractors` to extract the data from the XML element. Each `InfoExtractor` extracts one clearly identifiable piece of information from the XML element. Usually the data from the XML must be cleaned or mapped to other synonymous values. This is accomplished by chaining several `InfoExtractors`. The innermost gets the data from XML and returns it to the second extractor which further processes the data and returns it to the next and so on and finally the last extractor returns the data to `CollectedCardExtractor`. These chains are formed by giving the inner `InfoExtractor` object to the constructor of the outer `InfoExtractor`.



fi.helsinki.cs.orava.xml2rdf Class CollectedCardExtractor

java.lang.Object

└-fi.helsinki.cs.orava.xml2rdf.CollectedException

```
public class CollectedCardExtractor
extends java.lang.Object
```

`CollectedCardExtractor` is an `Extractor` that extracts data from XML Elements using several `InfoExtractors`. The `topicExtractor` is used to extract the RDF Resource itself. `dataExtractors` are then used to add properties to this Resource.

The extractors are given in the constructor, the resource extractor separately and others in a Map that has RDF Property objects as keys and `InfoExtractors` as values. The key defines the property used when adding the resource extracted by the value of the map. Note that there can only be one extractor for each property. You can overcome this limitation by using the `CombineInfoExtractor` when needed.

Fields

`dataExtractors`

```
protected java.util.Map dataExtractors
```

The data extractors. The keys of the Map are the Property objects used for adding the extracted information and values are the InfoExtractors used for extraction.

`topicExtractor`

```
protected fi.helsinki.cs.orava.xml2rdf.InfoExtractor topicExtractor
```

The InfoExtractor used to extract the Resource where properties can then be added with extractors in `dataExtractors`.

Constructors

`CollectedCardExtractor`

```
public CollectedCardExtractor(InfoExtractor topicExtractor,
                               java.util.Map dataExtractors)
```

Constructs a new `CollectedCardExtractor` which uses the given extractors.

Parameters:

`topicExtractor` - The `InfoExtractor` used to extract the resource where `dataExtractors` can add properties.

`dataExtractors` - The `InfoExtractors` used to extract the data for the Resource as a Map. The map has Property objects as keys, which represent the property used to add the extracted data. The values of the map are the `InfoExtractors` used to extract the data.

Methods

`extract`

```
public void extract(org.w3c.dom.Element element)
```

Performs extraction for one XML Element.

Parameters:

`element` - The element where data is extracted from.

fi.helsinki.cs.orava.xml2rdf Class CombineInfoExtractor

```
java.lang.Object
  |
  +-fi.helsinki.cs.orava.xml2rdf.InfoExtractor
      |
      +-fi.helsinki.cs.orava.xml2rdf.CombineInfoExtractor
```

```
public class CombineInfoExtractor
extends InfoExtractor
```

An `InfoExtractor` that combines the results of multiple `InfoExtractors` into one `Collection`.

Fields

chains

```
protected fi.helsinki.cs.orava.xml2rdf.InfoExtractor chains
```

Constructors

CombineInfoExtractor

```
public CombineInfoExtractor(InfoExtractor[] chains)
```

Creates a new `CombineInfoExtractor`.

Parameters:

`chains` - The `InfoExtractors` whose results are combined in extract.

Methods

extract

```
public java.util.Collection extract(java.util.Collection value)
```

`InfoExtractor` implementation. Returns a `Collection` containing all the results from calling the `extract` method of all `InfoExtractors` that were given to the constructor.

Parameters:

`value` - The value that is given to all the `InfoExtractors` whose results are to be combined.

Returns:

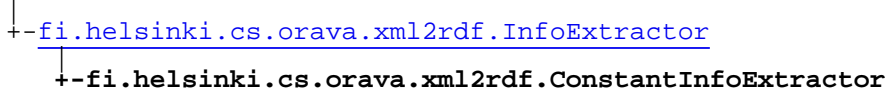
A `Collection` containing everything returned by all the `InfoExtractors` when their `extract` method was invoked.

extractSingle

```
public java.lang.Object extractSingle(java.lang.Object value)
```

fi.helsinki.cs.orava.xml2rdf Class ConstantInfoExtractor

java.lang.Object



```
public class ConstantInfoExtractor
extends InfoExtractor
```

An InfoExtractor that always returns the same Collection when extract is called.

Fields

constant

protected java.util.Collection **constant**

Constructors

ConstantInfoExtractor

```
public ConstantInfoExtractor(java.lang.Object node)
```

Creates a new ConstantInfoExtractor which will return a Collection containing the single object given as the parameter.

Parameters:

node - The object which will be returned by extract

ConstantInfoExtractor

```
public ConstantInfoExtractor(java.util.Collection constant)
```

Creates a new ConstantInfoExtractor which will return the Collection given as the parameter.

Parameters:

constant - The Collection which will be returned by extract

Methods

extract

```
public java.util.Collection extract(java.util.Collection value)
```

InfoExtractor implementation. Returns the Collection specified in the constructor.

Parameters:

value - Value where information should be extracted from but since this method always returns the same Collection this param is ignored.

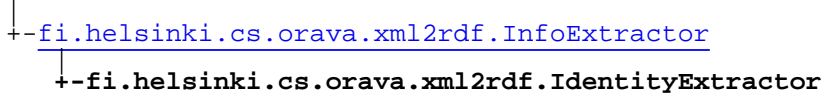
(continued from last page)

extractSingle

```
public java.lang.Object extractSingle(java.lang.Object value)
```

fi.helsinki.cs.orava.xml2rdf Class IdentityExtractor

java.lang.Object



```
public class IdentityExtractor
extends InfoExtractor
```

An `InfoExtractor` that returns the input verbatim and that does not use any chained `InfoExtractors`. This can be used as the first extractor in the chain.

Constructors

IdentityExtractor

```
public IdentityExtractor()
    Creates a new instance of IdentityExtractor
```

Methods

extractSingle

```
protected java.lang.Object extractSingle(java.lang.Object value)
```

extract

```
public java.util.Collection extract(java.util.Collection value)
```


fi.helsinki.cs.orava.xml2rdf

Class InfoExtractor

java.lang.Object

↳-fi.helsinki.cs.orava.xml2rdf.InfoExtractor

Direct Known Subclasses:

[TermCollectorExtractor](#), [CombineInfoExtractor](#), [LiteralMakerExtractor](#), [IdentityExtractor](#), [RegexInfoExtractor](#), [SplitInfoExtractor](#), [URIInfoExtractor](#), [ResourceMakerExtractor](#), [ConstantInfoExtractor](#), [XPathInfoExtractor](#), [MapInfoExtractor](#)

public abstract class **InfoExtractor**
extends java.lang.Object

`InfoExtractor` is a generic abstract base class for all `InfoExtractors`. It handles extractor chaining by requiring an inner extractor in the constructor.

If you are implementing an extractor that does not use chained extractors, you must override `extract` and must not call `extractMiddle`. If you do this, then it is safe to give `null` for the constructor as the chained extractor.

`InfoExtractor` has three generic (template) parameters: `O`, `M` and `I` for output type, intermediate type and input type respectively. It takes all its input as `I` and returns objects of type `O`. The intermediate type `M` is the type returned by the chained `InfoExtractor` and thus it is the type your implementation will actually get its parameters in.

Most implementations need only implement the abstract `extractSingle` method. If you override the `extract` method and your implementation doesn't call `extractSingle` or `super.extract`, then `extractSingle` is not needed and you may implement it by simply returning `null` there. If your implementation needs to return more than one value for each input value, then you must override `extract`. In this case you can use `extractMiddle` to perform the chained extraction.

Fields

chain

protected fi.helsinki.cs.orava.xml2rdf.InfoExtractor **chain**

The inner `InfoExtractor`.

Constructors

InfoExtractor

public **InfoExtractor**([InfoExtractor](#) chain)

Constructs a new `InfoExtractor` with the given inner extractor.

Parameters:

chain - The chained extractor that is first invoked when extraction is performed.

Methods

extract

public java.util.Collection **extract**(java.lang.Object value)

(continued from last page)

Performs the data extraction. This is a convenience method which calls `extract(Collection)` with a collection containing the single value given as parameter.

Parameters:

`value` - The object where information is extracted from.

Returns:

The extracted information.

extractMiddle

```
protected java.util.Collection extractMiddle(java.util.Collection value)
```

Calls the chained extractor and returns the `Collection` returned by it.

Parameters:

`value` - The objects where information is extracted from.

Returns:

The information extracted by the chained extractor.

extract

```
public java.util.Collection extract(java.util.Collection value)
```

Performs the data extraction. First invokes the chained extractor and then calls `extractSingle` for all values returned by the chained extractor. All non-null values returned by `extractSingle` are collected and then returned.

Parameters:

`value` - The objects where information is extracted from.

Returns:

The extracted information.

extractSingle

```
protected abstract java.lang.Object extractSingle(java.lang.Object value)
```

Extracts information from the given value.

Parameters:

`value` - The object where information is extracted from.

Returns:

The information extracted or `null` if `value` contains no information.

fi.helsinki.cs.orava.xml2rdf Class KlaffiExport

```
java.lang.Object
  |
  +-fi.helsinki.cs.orava.xml2rdf.KlaffiExport
```

```
public class KlaffiExport
extends java.lang.Object
```

The class that does data conversion for the Klaffi metadata. The `main` method first reads the needed ontologies then constructs an empty model where data can be added then constructs the `InfoExtractors` and the `CollectedCardExtractor` that perform the extraction. Then Reads files from the specified directory one at a time and does extraction for each and finally writes the extracted model into a file.

Constructors

KlaffiExport

```
public KlaffiExport()
```

Methods

main

```
public static void main(java.lang.String[] args)
throws java.lang.Exception
```

Performs the extraction.

Parameters:

`args` - Command line arguments. First is the directory containing the XML files to be processed. Second is the file name where the extracted data is written. Third is the directory containing all needed ontologies. Fourth is the file which contains URI mappings used in the extraction. Optional fifth argument is the file where term model is written.

makeExtractorMap

```
public static java.util.Map makeExtractorMap(java.lang.Object[] args)
```

Makes a Map that can be given as the second parameter to `CollectedCardExtractor` constructor. This is just a convenience method so that the extractor map can be done on one line.

Parameters:

`args` - Map contents. Arguments with even indices are the `Property` objects and arguments with odd indices are the `InfoExtractor` objects.

fi.helsinki.cs.orava.xml2rdf Class KlaffiExport.ProgramSerie

java.lang.Object

└-fi.helsinki.cs.orava.xml2rdf.KlaffiExport.ProgramSerie

static class **KlaffiExport.ProgramSerie**
extends java.lang.Object

Fields

count

java.lang.Integer **count**

urls

java.util.LinkedList **urls**

base_urls

java.util.LinkedList **base_urls**

exported

boolean **exported**

Constructors

KlaffiExport.ProgramSerie

KlaffiExport.ProgramSerie()

fi.helsinki.cs.orava.xml2rdf Class KlaffiXMLCleaner

java.lang.Object

└-fi.helsinki.cs.orava.xml2rdf.KlaffiXMLCleaner

```
public class KlaffiXMLCleaner
extends java.lang.Object
```

KlaffiXMLCleaner is a utility used to make invalid XML-files valid.

Constructors

KlaffiXMLCleaner

```
public KlaffiXMLCleaner()
```

Methods

main

```
public static void main(java.lang.String[] args)
throws java.lang.Exception
```

The main method for the utility. First argument must be the directory which contains the xml files to be fixed and second argument must be the directory where fixed files are written. These two directories cannot be same. Files from the source directory with file name suffixes xml and rdf are processed.

fi.helsinki.cs.orava.xml2rdf Class LiteralMakerExtractor

```
java.lang.Object
  |
  +-fi.helsinki.cs.orava.xml2rdf.InfoExtractor
    |
    +-fi.helsinki.cs.orava.xml2rdf.LiteralMakerExtractor
```

```
public class LiteralMakerExtractor
extends InfoExtractor
```

An InfoExtractor that converts a String into an RDF Literal.

Fields

model

```
protected com.hp.hpl.jena.rdf.model.Model model
```

Constructors

LiteralMakerExtractor

```
public LiteralMakerExtractor(com.hp.hpl.jena.rdf.model.Model model,
                             InfoExtractor chain)
```

Creates a new LiteralMakerExtractor.

Parameters:

model - The model used to create the returned Literals.

Methods

extractSingle

```
public com.hp.hpl.jena.rdf.model.Literal extractSingle(java.lang.String value)
```

Creates a Literal by calling model.createLiteral(value).

Parameters:

value - The String to be converted into a Literal.

Returns:

The created Literal.

fi.helsinki.cs.orava.xml2rdf Class MapInfoExtractor

```
java.lang.Object
  |
  +-fi.helsinki.cs.orava.xml2rdf.InfoExtractor
      |
      +-fi.helsinki.cs.orava.xml2rdf.MapInfoExtractor
```

```
public class MapInfoExtractor
extends InfoExtractor
```

An `InfoExtractor` that maps `Strings` into other `Strings` according to a predefined `Map`. If the map contains mappings `A->B` and `B->C` then when `A` is processed, it will result in `C`. You may form arbitrary long mapping sequences but you must not form loops, as they will make the extractor get stuck.

Fields

map

```
protected java.util.Map map
    The Map containing the String mappings.
```

Constructors

MapInfoExtractor

```
public MapInfoExtractor(java.lang.String file,
                        InfoExtractor chain)
```

Creates a new `MapInfoExtractor`.

Parameters:

`file` - The file where mappings are read from. See `parseMap` for details about the file format.
`chain` - The chained extractor.

MapInfoExtractor

```
public MapInfoExtractor(java.util.Map map,
                        InfoExtractor chain)
```

Creates a new `MapInfoExtractor`.

Parameters:

`map` - The `String` mappings.
`chain` - The chained extractor.

Methods

extractSingle

```
public java.lang.String extractSingle(java.lang.String value)
```

Maps a single `String` to another `String` as defined by the `Map` given to the constructor.

parseMap

```
public static java.util.Map parseMap(java.lang.String file)
    throws java.io.IOException
```

Reads a map from file, parses it and returns the parsed Map. See `parseMap(Reader)`.

parseMap

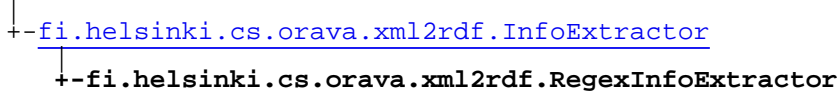
```
public static java.util.Map parseMap(java.io.Reader input)
    throws java.io.IOException
```

Reads a map from the provided `Reader` and returns the parsed Map. The input must contain one mapping per line where the key and the value are separated by any number of white space (or tab). Lines beginning with "#" are treated as comments and skipped.

fi.helsinki.cs.orava.xml2rdf

Class RegexInfoExtractor

java.lang.Object



```
public class RegexInfoExtractor
extends InfoExtractor
```

An `InfoExtractor` that performs regular expresion matching and/or replacing. The constructor is given a regular expression and a replacement `String`, both using the syntax of `java.util.regex.Pattern`. Optionally a boolean may be given indicating whether the input must match the regular expression for it to be returned, this is `false` if not given. If you do not want to perform any replacing, make a capturing group in your regular expression and use "\$1" as replacement `String`. See `extractSingle` and `java.util.regex.Pattern`.

Fields

pattern

```
protected java.util.regex.Pattern pattern
    The compiled regular expression pattern.
```

replace

```
protected java.lang.String replace
    The replacement String
```

mustMatch

```
protected boolean mustMatch
    Whether a match is required for the input to be passed forward.
```

Constructors

RegexInfoExtractor

```
public RegexInfoExtractor(java.lang.String regex,
                           java.lang.String replace,
                           boolean mustMatch,
                           InfoExtractor chain)
```

Creates a new `RegexInfoExtractor`.

Parameters:

- `regex` - The regular expression used.
- `replace` - The replacement `String`.
- `mustMatch` - Whether a match is required for the input to be passed forward.
- `chain` - The chained extractor.

(continued from last page)

RegexInfoExtractor

```
public RegexInfoExtractor(java.lang.String regex,  
                           java.lang.String replace,  
                           InfoExtractor chain)
```

Creates a new `RegexInfoExtractor`. The constructed object will not require a match for the input to be passed forward.

Parameters:

- `regex` - The regular expression used.
- `replace` - The replacement `String`.
- `chain` - The chained extractor.

Methods

extractSingle

```
public java.lang.String extractSingle(java.lang.String value)
```

Processes a single `String` using the predefined regular expression and replacement `String`. If the `mustMatch` parameter in the constructor was `true` and the input doesn't contain a region matching the regular expression, then `null` is returned. Otherwise all occurrences that match the regular expression are replaced with the replacement `String` and the resulting `String` is returned.

Parameters:

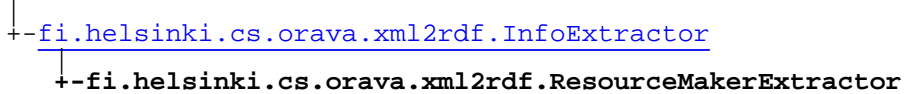
- `value` - The `String` used as input.

Returns:

A `String` with the replacement performed or `null` as described above.

fi.helsinki.cs.orava.xml2rdf Class ResourceMakerExtractor

java.lang.Object



```
public class ResourceMakerExtractor
extends InfoExtractor
```

An `InfoExtractor` that makes `Resources` from `URI Strings`. A special term `Model` may be provided which may contain information about synonymous terms and where information about unknown terms (that is terms not already in the term model) is added. Synonym properties must be defined using the `SYNONYMOF_URI` as the `URI` of the property. Unknown terms are marked with type `UNKNOWN_URI` when added to the term model.

`InfoExtractor` may also be constructed in a mode that does not allow same resource to be extracted twice, that is no two resources with same `URI` may be extracted. If this happens in this mode, a warning is written to `System.stderr` and the resource with the `URI` is removed from the model and no resources with that `URI` are extracted in the future and `extractSingle` will thus return `null` in such cases.

Fields

URI_PREFIX

```
public static final java.lang.String URI_PREFIX
    Prefix for other URIs.
```

SYNONYMOF_URI

```
public static final java.lang.String SYNONYMOF_URI
    URI for synonym property. If A is synonym of B then B is used as term instead of A, when A is extracted from XML.
    Synonyms, may form arbitrary long chains although it is recommended that they remain only a couple of statements long.
    Also, synonymof- statements must not form loops and if a resource is marked as the synonym of multiple resources, one of
    them is used randomly.
```

UNKNOWN_URI

```
public static final java.lang.String UNKNOWN_URI
    URI for a class used to mark terms, that were not found in the term model and were thus created during extraction process.
```

cardModel

```
protected com.hp.hpl.jena.rdf.model.Model cardModel
    The model used to create Resources returned by extractSingle.
```

termModel

```
protected com.hp.hpl.jena.rdf.model.Model termModel
    The term model used for synonym lookups and unknown term logkeeping.
```

blockedURIs

protected java.util.HashSet **blockedURIs**

URIs that are not allowed to create resources. Duplicate URIs will be added here if `noDuplications` is true.

noDuplications

protected boolean **noDuplications**

If true, it is considered an error if same resource is extracted twice.

Constructors

ResourceMakerExtractor

```
public ResourceMakerExtractor(com.hp.hpl.jena.rdf.model.Model cardModel,
                              InfoExtractor chain)
```

Creates a new ResourceMakerExtractor without a term model.

Parameters:

`cardModel` - The model used to create Resources returned by `extractSingle`.
`chain` - The chained extractor.

ResourceMakerExtractor

```
public ResourceMakerExtractor(com.hp.hpl.jena.rdf.model.Model cardModel,
                              boolean noDuplications,
                              InfoExtractor chain)
```

Creates a new ResourceMakerExtractor without a term model.

Parameters:

`cardModel` - The model used to create Resources returned by `extractSingle`.
`noDuplications` - Whether same resource may be extracted twice.
`chain` - The chained extractor.

ResourceMakerExtractor

```
public ResourceMakerExtractor(com.hp.hpl.jena.rdf.model.Model cardModel,
                              com.hp.hpl.jena.rdf.model.Model termModel,
                              InfoExtractor chain)
```

Creates a new ResourceMakerExtractor with a term model.

Parameters:

`cardModel` - The model used to create Resources returned by `extractSingle`.
`termModel` - The term model used for synonym lookups and unknown term logkeeping.
`chain` - The chained extractor.

ResourceMakerExtractor

```
public ResourceMakerExtractor(com.hp.hpl.jena.rdf.model.Model cardModel,
                              com.hp.hpl.jena.rdf.model.Model termModel,
                              boolean noDuplications,
                              InfoExtractor chain)
```

Creates a new ResourceMakerExtractor with a term model.

Parameters:

`cardModel` - The model used to create Resources returned by `extractSingle`.
`termModel` - The term model used for synonym lookups and unknown term logkeeping.

(continued from last page)

`noDuplicates` - Whether same resource may be extracted twice.

`chain` - The chained extractor.

Methods

extractSingle

```
public com.hp.hpl.jena.rdf.model.Resource extractSingle(java.lang.String value)
```

`InfoExtractor` implementation. Makes a `Resource` from the given URI String. If a term model was given to the constructor, performs synonym lookup and possibly adds data about the previously unknown term.

fi.helsinki.cs.orava.xml2rdf

Class SplitInfoExtractor

```
java.lang.Object
├── fi.helsinki.cs.orava.xml2rdf.InfoExtractor
│   └── fi.helsinki.cs.orava.xml2rdf.SplitInfoExtractor
```

```
public class SplitInfoExtractor
extends InfoExtractor
```

An `InfoExtractor` that splits the input `String` into multiple `Strings` using a regular expression delimiter.

Fields

splitPattern

```
protected java.util.regex.Pattern splitPattern
```

The compiled regular expression used to split input `Strings`.

Constructors

SplitInfoExtractor

```
public SplitInfoExtractor(java.lang.String splitRegex,
                          InfoExtractor chain)
```

Creates a new `SplitInfoExtractor`.

Parameters:

`splitRegex` - The regular expression that is used to find the split positions.
`chain` - The chained extractor.

Methods

extract

```
public java.util.Collection extract(java.util.Collection value)
```

Overridden `InfoExtractor` method. Calls the chained extractor and then splits all its outputs, combines them into a single `Collection` and returns it.

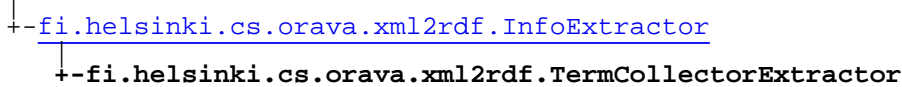
extractSingle

```
public java.lang.String extractSingle(java.lang.String value)
```

`InfoExtractor` implementation. Does nothing and returns `null` as it is not needed because `extract` is overridden.

fi.helsinki.cs.orava.xml2rdf Class TermCollectorExtractor

java.lang.Object



public class **TermCollectorExtractor**
extends [InfoExtractor](#)

An `InfoExtractor` that collects unknown terms into a `Model` and prefills properties for them. This is achieved by using two extractor chains, the first is the normal extractor chain which is expected to return a `String` suitable for RDF resource label. The second chain gets this `String` and is expected to return a `URI` that will be used to check whether the extracted term is unknown or not and if necessary used to create a new resource. It is also the value returned by the `extract` method.

This class uses two `Models`. One is used to lookup which resource is known and which is unknown. If the `Model` contains a resource with the `URI` returned by the second extractor chain, then the term is interpreted to be known and otherwise it is unknown. If the term is unknown, a new resource is created in the other `Model` and it's label is set to the value returned by the first extractor chain. Note that these two `Models` may be set to be the same `Model`. You may also use `null` for both models in which case `URIs` are not checked and no new resources are created, the `extract` method only returns the `URIs` returned by the second extractor chain.

Fields

unknownModel

```
private com.hp.hpl.jena.rdf.model.Model unknownModel
```

knownModel

```
private com.hp.hpl.jena.rdf.model.Model knownModel
```

uriExtractor

```
private fi.helsinki.cs.orava.xml2rdf.InfoExtractor uriExtractor
```

Constructors

TermCollectorExtractor

```
public TermCollectorExtractor(com.hp.hpl.jena.rdf.model.Model knownModel,
                             com.hp.hpl.jena.rdf.model.Model unknownModel,
                             InfoExtractor chain,
                             InfoExtractor uriExtractor)
```

Creates a new instance of `TermCollectorExtractor`.

Parameters:

`knownModel` - The model containing the known terms.

(continued from last page)

`unknownModel` - The model where unknown terms are added.

`chain` - The extractor chain that must return a value which is suitable for label of RDF resources and which is also used as the input to the second extractor chain.

`uriExtractor` - The extractor chain used to change the output of the first extractor chain into a URI.

Methods

extractSingle

```
public java.lang.String extractSingle(java.lang.String value)
```

InfoExtractor implementation. This class overrides the `extract` method so this always returns null.

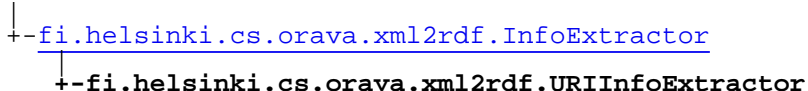
extract

```
public java.util.Collection extract(java.util.Collection valueOuter)
```

InfoExtractor implementation. Performs the extraction.

fi.helsinki.cs.orava.xml2rdf Class URIInfoExtractor

java.lang.Object



```
public class URIInfoExtractor
extends InfoExtractor
```

An `InfoExtractor` that makes a `URI String` from an extracted `String`. This is done by removing or changing special characters in the extracted `String` into other characters and then appending the result after a provided `URI prefix`.

Fields

uriPrefix

```
protected java.lang.String uriPrefix
```

The prefix used when creating URIs.

Constructors

URIInfoExtractor

```
public URIInfoExtractor(java.lang.String uriPrefix,
                        InfoExtractor chain)
```

Creates a new `URIInfoExtractor`.

Parameters:

`uriPrefix` - The prefix used when creating URIs. This will usually end in "#". #param chain The chained extractor.

Methods

extractSingle

```
public java.lang.String extractSingle(java.lang.String value)
```

Performs the `URI conversion` using `makeURI` method.

Parameters:

`value` - The extracted `String` that is converted into a `URI`.

Returns:

The created `URI`.

makeURI

```
protected java.lang.String makeURI(java.lang.String extracted)
```

Performs the `URI conversion` by converting special non `ASCII` characters into `ASCII` equivalents or underscore ("`_`") and then appending the resulting `String` after the `URI prefix` given to the constructor.

(continued from last page)

Parameters:

extracted - The String that is converted into a URI.

Returns:

The created URI.

fi.helsinki.cs.orava.xml2rdf

Class XMLTools

```
java.lang.Object
  |
  +-fi.helsinki.cs.orava.xml2rdf.XMLTools
```

```
public class XMLTools
extends java.lang.Object
```

XML centric utility methods.

Constructors

XMLTools

```
public XMLTools()
```

Methods

getNodeContents

```
public static java.lang.String getNodeContents(org.w3c.dom.Node n)
```

Returns the textual contents of the given XML Node. The returned String will not contain element tags, attributes or comments only the textual contents of the given Node

Parameters:

n - The Node whose contents are to be returned.

Returns:

The extracted contents.

getNodeContents

```
private static void getNodeContents(org.w3c.dom.Node n,
java.lang.StringBuffer sb)
```

Same as getNodeContents(Node) but appends the results to the given StringBuffer.

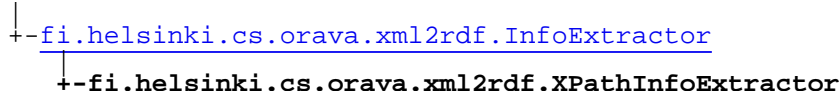
Parameters:

n - The Node whose contents are to be extracted.

sb - The StringBuffer where text is appended.

fi.helsinki.cs.orava.xml2rdf Class XPathInfoExtractor

java.lang.Object



public class **XPathInfoExtractor**
extends [InfoExtractor](#)

An `InfoExtractor` that extracts information from an XML Element using an XPath query. The expression used in the query is given to the constructor. The `extract` method will return a `Collection` of `Strings` from the results of the XPath query. This `Collection` may contain any number of elements, including zero.

Fields

xPathExpression

protected javax.xml.xpath.XPathExpression **xPathExpression**

The compile XPathExpression.

Constructors

XPathInfoExtractor

public **XPathInfoExtractor**(java.lang.String path)

Creates a new `XPathInfoExtractor`. #param path The XPath expression used to extract information.

Methods

extract

public java.util.Collection **extract**(java.util.Collection elements)

Overridden `InfoExtractor` method that performs the extraction using the provided XPath expression.

Parameters:

elements - The XML Elements where information is extracted from.

Returns:

The results of the queries converted into a `Collection` of `Strings`.

extractSingle

public java.lang.String **extractSingle**(org.w3c.dom.Element element)

Index

B

base_urls 12
blockedURIs 20

C

cardModel 19
chain 9
chains 5
CollectedCardExtractor 4
CombineInfoExtractor 5
constant 6
ConstantInfoExtractor 6
count 12

D

dataExtractors 4

E

exported 12
extract 4, 5, 6, 8, 9, 10, 22, 24, 28
extractMiddle 10
extractSingle 5, 6, 8, 10, 14, 15, 18, 21, 22, 24, 25, 28

G

getNodeContents 27

I

IdentityExtractor 8
InfoExtractor 9

K

KlaffiExport 11
KlaffiXMLCleaner 13
knownModel 23

L

LiteralMakerExtractor 14

M

main 11, 13
makeExtractorMap 11
makeURI 25
map 15
MapInfoExtractor 15
model 14
mustMatch 17

N

noDuplicates 20

P

parseMap 16
pattern 17
ProgramSerie 12

R

RegexInfoExtractor 17
replace 17
ResourceMakerExtractor 20

S

SplitInfoExtractor 22
splitPattern 22
SYNONYMOF_URI 19

T

TermCollectorExtractor 23
termModel 19
topicExtractor 4

U

UNKNOWN_URI 19
unknownModel 23

URI_PREFIX 19
uriExtractor 23
URIInfoExtractor 25
uriPrefix 25
urls 12

X

XMLTools 27
xpathExpression 28
XPathInfoExtractor 28