

The TaxonClassifierLib API

Calling TaxonClassifierLib

You can call the TaxonClassifierLib from any language on any operating system that supports Windows DLLs.

There are 3 public functions:

```
makeLookupTaxonomy
classify
classifyText
```

makeLookupTaxonomy

makeLookupTaxonomy converts the original taxonomy into a lookup taxonomy based on the terms in the taxonomy.

A call looks like:

```
string lookupTaxonomy = makeLookupTaxonomy(taxonomy)
```

where taxonomy is a JSON string with the original taxonomy and lookupTaxonomy is a string with the resulting lookup taxonomy.

classify and classifyText

classifyText is a wrapper for classify, so that the calling program avoids using the Newtonsoft JObject.

A call to classify looks like:

```
JObject result = classify(text, lookupTaxonomy, settings)
```

where text is the string with the text to analyse, lookupTaxonomy is the JObject with the lookup taxonomy and settings is a JObject with settings (see below).

The result is a JObject with the result or errors.

A call to classifyText looks like:

```
string resultJSON = classifyText(text, lookupTaxonomyJSON, settings)
```

where text is the string with the text to analyse, lookupTaxonomyJSON is the text string with the lookup taxonomy and settings is a Dictionary<string,string> with settings (see below).

The resultJSON is a string with the result or errors in JSON.

settings

numberResultsReturned, a number equal to or greater than 0. 0 means all.

ignoreTermConstraints, either 0 or 1.

0 means use the full classification.

1 ignores the requireTerm, excludeOnTerm, requireClass, excludeOnClass as well as the class thresholds.

onNoResultsIgnoreTermConstraints, either 0 or 1.

0 If a full classification returned 0 results do nothing.

1 If a full classification returned 0 results another classification is run with the ignoreTermConstraints.

onNoResultsUseAlternativeTaxonomy, either 0 or 1.

0 If a full classification returned 0 results do nothing.

1 If a full classification returned 0 results another classification is run with the onNoResultsUseAlternativeTaxonomy.

returnShortResult, either 0 or 1.

0 Return the full result.

1 Return short result, i.e. only ID and title.

Return value

A JSON string with the status of the classification and the found classes if any.

```
{
  "status":
  {
    "status":["OK","Error"]
    "errors":
    {
      0:
      {
        "errorCode":[0-9]+,
        "errorText":<STRING>
      },
      ...
    },
    "information":
    {
      0:
      {
        "informationCode":[0-9]+,
        "informationText":<STRING>
      },
      ...
    }
  }
}
```

```

    }
  },
  "classes":
  {
    "[0-9\\.]+":
    {
      "title": <STRING>,
      "hidden": [0|1],
      "exclusive": [0|1],
      "superClass": [0|1],
      "thresholdWeight": [0-9]+,
      "thresholdCount": [0-9]+,
      "thresholdCountUnique": [0-9]+,
      "requireClass": <STRING>,
      "excludeOnClass": <STRING>,
      "includeClass": <STRING>,
      "terms":
      {
        <STRING>:
        {
          "scoreWeight": [0-9]+,
          "scoreCount": [0-9]+,
          "scoreFirstPosition": [0-9]+,
          "hits":
          {
            <STRING>: [0-9]+
            ...
          },
          "required": [0|1],
          "requiredOr": [0|1],
          "requireText": <STRING>,
          "excludeOnText": <STRING>,
          "requireTerm": <STRING>,
          "excludeOnTerm": <STRING>,
          "requireClass": <STRING>,
          "excludeOnClass": <STRING>,
          "forceIncludeClass": [0|1],
          "forceExcludeClass": [0|1],

```

```

        "forceSuperClass":[0|1]
    }
},
"classificationMethod":<STRING>,
"scoreCount":[0-9]+,
"scoreWeight":[0-9]+,
"scoreFirstPosition":[0-9]+,
"scoreFirstPositionExtraWeight":[0-9]+,
"scoreTotal":[0-9]+,
"scoreConfidenceCoefficient":[0,100]
}
}
}
}
}

```

"status"

"status":<STRING>

The "status" value is always present and holds either "OK" or "Error".

If the "status" is "OK" the classification went okay. There might be additional information in the "information" section.

If the "status" is "error" one or more errors occurred and they are described in the "errors" section.

See error and information codes in the "Error and Information codes" section later.

"classes"

[0-9.]+

is the class ID.

Contains numbers and possibly ".".

Examples:

1

01

01.02.03

"title":<STRING>

The class title without the numeric ID.

Contains characters.

Examples:

EU countries

Asia

Africa

"exclusive":[0|1]

Whether the class is exclusive in the result.

If the class is exclusive it is only returned if no other class with the same parent is returned.

Contains a number. The number can be 0 or 1.

Default: 0. The default is used in various places, i.e. in the classification.

Examples:

0

1

"hidden":[0|1]

Whether the class is hidden in the result.

This information is only useful if either of the settings ignoreTermConstraints or onNoResultsIgnoreTermConstraints is set otherwise it is always 0.

Contains a number. The number can be 0 or 1.

Default: 0. The default is used in various places, i.e. in the classification.

Examples:

0

1

"superClass":[0|1]

Whether the class is a superClass in the result.

If there are one or more super classes only the super classes are returned in the result.

Contains a number. The number can be 0 or 1.

Default: 0. The default is used in various places, i.e. in the classification.

Examples:

0

1

"thresholdWeight":[0-9]+

The threshold weight of the class.

The weight of the class is the sum of the weights of each term in class found in the text.

The weight of a term is calculated by multiplying the number of hits by the weight of the term.

The sum of the weights of the terms must be greater or equal to the thresholdWeight for the class to be included in the result.

Contains a number. The number can be negative, 0 or positive.

Default: 5. The default is used in various places, i.e. in the classification.

Examples:

5

10

"thresholdCount":[0-9]+

The threshold count of the class.

The count of the class is the sum of the hits of each term in the class found in the text.

The sum of the hits of the terms must be greater or equal to the thresholdCount for the class to be included in the result.

Contains a number. The number can be 0 or positive.

Default: 1. The default is used in various places, i.e. in the classification.

Examples:

1

10

"thresholdCountUnique":[0-9]+

The threshold count of unique terms in the class.

The count of unique terms in the class is the sum of the hits of unique terms in the class found in the text.

The count of the hits of the terms must be greater or equal to the thresholdCountUnique for the class to be included in the result.

Contains a number. The number can be 0 or positive.

Default: 1. The default is used in various places, i.e. in the classification.

Examples:

1

2

"requireClass":<STRING>

A list of class ids required for this class to be included in result.

Contains a string. The class ids are separated by "||" for logical OR and "&&" for logical AND. The operators are evaluated from left to right.

Default: "". The default is used in various places, i.e. in the classification.

Examples:

01

01||02 (means "(01 OR 02)")

01&&02||03 (means "(01 AND 02) OR 03")

"excludeOnClass":<STRING>

A list of class ids which causes this class to be excluded from result.

Contains a string. The class ids are separated by "||" for logical OR and "&&" for logical AND. The operators are evaluated from left to right.

Default: "". The default is used in various places, i.e. in the classification.

Examples:

01

01||02 (means "(01 OR 02)")

01&&02||03 (means "(01 AND 02) OR 03")

"includeClass":<STRING>

A list of class ids which terms are include into this class' terms before classification.

If you have several classes that contains the same list of terms, e.g. names or locations,

it can be easier to maintain if the terms are moved to an new class and that class is included in the other classes. This way you only have to administer the term in one place.

Contains a string. The class ids are separated by "|".

Default: "". The default is used in various places, i.e. in the classification.

Examples:

01

01|02

01|02|03

"terms": {}

An array of terms, see below.

"title": "<STRING>"

The title of the term from the taxonomy. It is also the key for the array of both calculated scores and informational settings from the taxonomy.

Contains characters.

Examples:

Denmark

Spain

"scoreWeight":[0-9]+

The total weight of the term calculated by the weight of the term multiplied by the number of hits.

Contains a number. The number can be 1 or higher.

Examples:

5

1000

"scoreCount":[0-9]+

The total count of hits of the term and all it's forms (with prefixes and suffixes). This number is used to calculate the weight (see above).

Contains a number. The number can be 1 or higher.

Examples:

1

2

"scoreFirstPosition":[0-9]+

The position in the text where the term (or one of it's forms) was first read. This number is used in the class' scoreFirstPosition.

Contains a number. The number can be 1 or higher.

Examples:

1

739

"hits": array of terms

The hits array contains a list of the term and its forms (prefixes and suffixes) and the number of hits for each on the form:

```
{  
"<STRING>":[0-9]+  
}
```

Contains an array. The size of the array is 1 or higher.

Examples:

```
{  
"Employment":2,  
"Unemployment":1  
}
```

"required":[0|1]

Whether the term is required taken from the taxonomy.

The term must appear at least once in the text for the class to fire.

Contains a number. The number can be 0 or 1.

Default: 0. The default is used in various places, i.e. in the classification.

Examples:

0

1

"requiredOr":[0|1]

Whether the term is requiredOr taken from the taxonomy.

At least one of the terms with requiredOr set to 1 must appear at least once in the text for the class to fire.

Contains a number. The number can be 0 or 1.

Default: 0. The default is used in various places, i.e. in the classification.

Examples:

0

1

"requireText":"<STRING>"

The required term(s) from the taxonomy.

At least one of the required terms must appear at least once in the text for the term to fire.

The terms may or may not be a term in this or other classes.

Contains a list of strings separated by || or &&. The string can be empty. The strings in the list will be trimmed.

Default: "". The default is used in various places, i.e. in the classification.

Examples:

term

term1||term2&&term3. This requires either term1 or term2 to be present and term3 must be

present in the text.

/date\s+([0-3]?[0-9][\.\V\-\-][0-1]?[0-9][\.\V\-\-s][12][0-9]{3})/ with the text 'Date 1.1 2013' results in a hit.

"excludeOnText": "<STRING>"

The excluding term(s) from the taxonomy.

If least one of the excluding terms appear at least once in the text the term is excluded.

The terms may or may not be a term in this or other classes.

Contains a list of strings separated by || or &&. The string can be empty. The strings in the list will be trim'ed.

Default: "". The default is used in various places, i.e. in the classification.

Examples:

term

term1||term2&&term3. The term will be excluded if either term1 or term2 is present and term3 is present in the text.

/date\s+([0-3]?[0-9][\.\V\-\-][0-1]?[0-9][\.\V\-\-s][12][0-9]{3})/ with the text 'Date 1.1 2013' results in the term being excluded.

"requireTerm": "<STRING>"

The required term(s) from the taxonomy.

The required terms are terms from the same class in the result.

Contains a list of strings separated by || or &&. The string can be empty. The strings in the list will be trimmed.

Default: "". The default is used in various places, i.e. in the classification.

Examples:

term

term1||term2&&term3. This requires either term1 or term2 to be present and term3 must be present in the text.

"excludeOnTerm": "<STRING>"

The excluding term(s) from the taxonomy.

The excluding terms are terms from the same class in the result.

Contains a list of strings separated by || or &&. The string can be empty. The strings in the list will be trimmed.

Default: "". The default is used in various places, i.e. in the classification.

Examples:

term

term1||term2&&term3. The term will be excluded if either term1 or term2 is present and term3 is present in the result.

"requireClass": "<STRING>"

The required class(es) from the taxonomy.

The required class must also be fired for this term to be fired. This is used on more common terms that might be used in different contexts, e.g. Jaguar is both a car, an animal and a operating system.

The required class can be another class or the same class as the term belongs to. If the

required class is the same as the class of the term, another term in the class is required for the term to be fired.

Contains a list of class IDs separated by || or &&. The string can be empty.

Default: "". The default is used in various places, i.e. in the classification.

Examples:

01

01.02.03

01||02&&03. The term will be included if either of the classes 01 or 02 is present and 03 is in

the results.

"excludeOnClass": "<STRING>"

The excluding class(es) from the taxonomy.

If the excluding is fired this class is removed from the result.

Contains a list of class IDs separated by || or &&. The string can be empty.

Default: "". The default is used in various places, i.e. in the classification.

Examples:

01

01.02.03

01||02&&03. The term will be excluded if either of the classes 01 or 02 is present and 03 is in the results.

"forceIncludeClass": [0|1]

If the term is hit then force the class to be included in the result disregarding all other term and class constraints.

Contains an integer.

Default: 0. The default is used in various places, i.e. in the classification.

Examples:

0

1

"forceExcludeClass": [0|1]

If the term is hit then force the class to be excluded in the result disregarding all other term and class constraints.

Contains an integer.

Default: 0. The default is used in various places, i.e. in the classification.

Examples:

0

1

"forceSuperClass": [0|1]

If the term is hit then force the class to be a superClass in the result disregarding all other term and class constraints.

Contains an integer.

Default: 0. The default is used in various places, i.e. in the classification.

Examples:

0

1

Back to the class settings.

"classificationMethod": "<STRING>"

The classificationMethod is the type of classification that was used to produce the result.

Contains a string. The string can be "Full classification", "Alternative taxonomy",

"IgnoreTermConstraints" or "onNoResultsIgnoreTermConstraints".

Examples:

"Full classification"

"IgnoreTermConstraints"

"scoreCount": [0-9]+

The number of total number of hits on terms in the class.

Contains a number. The number can be 1 or higher.

Examples:

1

7

"scoreWeight": [0-9]+

The total weight of all the hits in the class.

Contains a number. The number can be 1 or higher.

Examples:

5

1000

"scoreFirstPosition": [0-9]+

The position of the first found term in the class. It is found from the terms' scoreFirstPosition setting.

Contains a number. The number can be 1 or higher.

Examples:

1

378

"scoreFirstPositionExtraWeight": [0-9]+

If the first found term belongs to this class an additional weight of 10 is added to the class weight.

Contains a number. The number can be 0 or 10.

Examples:

0

10

"scoreTotal": [0-9]+

The sum of the scoreWeight and the scoreFirstPositionExtraWeight gives the total score for the class.

Contains a number. The number can be 1 or higher.

Examples:

5

1010

"scoreConfidenceCoefficient":0-100

The scoreConfidenceCoefficient is a relative measure of the accuracy of the result. The scoreConfidenceCoefficient is calculated as the weight of the top result minus the weight of the runner-up result divided by the weight of the top result.

Together with the scoreWeight it provides an indication of how certain Taxon is of the accuracy of the result.

A value of 0 means that the top two results have the same weight and we do not know which is the best. Usually the 100 is only achieved when there is just one class in the result.

Contains a percentage. The number can be 0 to 100.

Examples:

0

56

100