
# Viglet Turing ES
## *Connectors*

Viglet Team

Version 0.3.10, 25-12-2024

# Table of Content

# Preface

There are several connectors to allow you to index content in Viglet Turing ES.

# Chapter 1. Apache Nutch

Plugin for Apache Nutch to index content using crawler.

## 1.1. Installation

Turing support Apache Nutch 1.12 and 1.8 only, so go to https://viglet.com/turing/download/ and click on "Integration > Apache Nutch" link to download the turing-nutch-<NUTCH_RELEASE>-bin.zip.

1. Extract turing-nutch-<NUTCH_RELEASE>-bin.zip file into /appl/viglet/turing/nutch.

```
mkdir -p /appl/viglet/turing/nutch
unzip turing-nutch.zip -d /appl/viglet/turing/nutch
```

2. Download and install Apache Nutch 1.12 or 1.18 binary into http://nutch.apache.org > Downloads > apache-nutch-<NUTCH_RELEASE>-bin.tar.gz.

```
mkdir -p /appl/apache/
cp apache-nutch-<NUTCH_RELEASE>-bin.tar.gz /appl/apache
cd /appl/apache
tar -xvzf apache-nutch-<NUTCH_RELEASE>-bin.tar.gz
ln -s apache-nutch-<NUTCH_RELEASE>-bin nutch
```

3. Copy the Turing Plugin to Apache Nutch.

```
cp -R /appl/viglet/turing/nutch/indexer-viglet-turing /appl/apache/nutch/plugins
cp -f /appl/viglet/turing/nutch/conf/* /appl/apache/nutch/conf/
```

## 1.2. Configuration

### 1.2.1. Nutch 1.12

This step is only for Apache Nutch 1.12. Edit the /appl/apache/nutch/conf/nutch-site.xml, add or modify the following properties:

```
<property>
  <name>solr.server.url</name>
  <value>http://127.0.0.1:2700/Sample</value>
  <description>
      Turing URL + "/" + Turing Semantic Navigation Site.
  </description>
</property>
<property>
  <name>turing.url</name>
  <value>http://127.0.0.1:2700</value>
  <description>
      Defines the Turing URL into which data should be indexed using the
      indexer-turing plugin.
  </description>
</property>
<property>
  <name>turing.site</name>
  <value>Sample</value>
  <description>
      Defines the Turing Semantic Navigation Site.
  </description>
</property>
<property>
  <name>turing.auth</name>
  <value>true</value>
  <description>
      Whether to enable HTTP basic authentication for communicating with Turing.
Use the username and password properties to configure your credentials.
  </description>
</property>
<property>
  <name>turing.username</name>
  <value>admin</value>
  <description>
      The username of Turing server.
  </description>
</property>
```

```
<property>
  <name>turing.password</name>
  <value>admin</value>
  <description>
      The password of Turing server.
  </description>
</property>
<property>
  <name>turing.timestamp.field</name>
  <value>modification_date</value>
  <description>
      Field used to store the timestamp of indexing. The default value is "tstamp".
  </description>
</property>
<property>
  <name>turing.field.type</name>
  <value>Page</value>
  <description>
      Type of Content. The default value is "Page".
  </description>
</property>
<property>
  <name>turing.field.source_appS</name>
  <value>Nutch</value>
  <description>
      Name of Source Application. The default value is "Nutch".
  </description>
</property>
<!--
<property>
  <name>turing.field.hello</name>
  <value>foo</value>
  <description>
      This a test.
  </description>
</property>
<property>
  <name>turing.field.world</name>
  <value>bar</value>
  <description>
      This is another test.
  </description>
</property>
-->
```

If you want to add metatag values, make sure parse-metatags is set in plugin.includes and

add the following parameters:

```
<property>
    <name>metatags.names</name>
    <value>*</value>
    <description> Names of the metatags to extract, separated by ','.
  Use '*' to extract all metatags. Prefixes the names with 'metatag.'
  in the parse-metadata. For instance to index description and keywords,
  you need to activate the plugin index-metadata and set the value of the
  parameter 'index.parse.md' to 'metatag.description,metatag.keywords'.
</description>
  </property>


  <property>
    <name>index.parse.md</name>
    <value>metatag.description,metatag.keywords,metatag.language</value>
    <description>
  Comma-separated list of keys to be taken from the parse metadata to generate
fields.
  Can be used e.g. for 'description' or 'keywords' provided that these values are
generated
  by a parser (see parse-metatags plugin)
  </description>
  </property>

  <property>
    <name>http.content.limit</name>
    <value>6553600</value>
  </property>
```

**turing.xml File**

The plugin uses /appl/apache/nutch/conf/turing-mapping.xml to perform the actions:

1. Rename the fields using, for example: `<field source =" content "dest =" text "/>` where the `source` attribute is the original field name and the ` dest` attribute is the new attribute name.
2. Dynamically add the semantic navigation site name, based on the page URL, for example: `<site url="https://viglet.com" snSite="Sample"/>`, where the `url` attribute is the URL prefix and the `snSite` attribute is the semantic navigation site name that was configured in the Turing console.

3. Defines the attribute which is the unique key that will be used when indexing in Turing semantic navigation, for example: `<uniqueKey>id</uniqueKey>`, where the value into `uniqueKey` tag is the attribute.

```
<mapping>
  <fields>
    <field source="content" dest="text"/>
    <field source="title" dest="title"/>
    <field source="host" dest="host"/>
    <field source="segment" dest="segment"/>
    <field source="boost" dest="boost" remove="true"/>
    <field source="digest" dest="digest"/>
    <field source="tstamp" dest="tstamp"/>
    <field source="metatag.description" dest="description" />
  </fields>
  <sites>
    <site url="https://viglet.com" snSite="Sample"/>
  </sites>
  <uniqueKey>id</uniqueKey>
</mapping>
```

#### Field with Timestamp

Can specify what is the field will be used to store the timestamp of indexing. The default value is `tstamp`. So modify the value of `turing.timestamp.field` property into nutch-site.xml:

```
<property>
  <name>turing.timestamp.field</name>
  <value>modification_date</value>
  <description>
      Field used to store the timestamp of indexing. The default value is "tstamp".
  </description>
</property>
```

#### Source App Name

Turing ES Semantic Navigation Site allows to index content from many sources, so can identify where the content was indexed, can specify the name of the source changing the `turing.field.source_apps` into nutch-site.xml file. The default value is `Nutch`:

```
<property>
  <name>turing.field.source_apps</name>
  <value>Nutch</value>
  <description>
      Name of Source Application. The default value is "Nutch".
  </description>
</property>
```

#### Fixed Fields

To create new fixed field during indexing, add new properties with prefix `turing.field` + `name of new custom field` into nutch-site.xml file, for example:

```
<property>
  <name>turing.field.hello</name>
  <value>foo</value>
  <description>
      This a test.
  </description>
</property>
<property>
  <name>turing.field.world</name>
  <value>bar</value>
  <description>
      This is another test.
  </description>
</property>
```

| | |
|---|---|
| **IMPORTANT** | Need add these fields to Solr schema.xml file and create them in Semantic Navigation Site > Fields |

#### Parameters

Modify the following parameters:

*Table 1. nutch-site.xml parameters*

| Parameter | Description | Default value |
|---|---|---|
| solr.server.url | Turing URL + "/" + Turing Semantic Navigation Site. | - |
| turing.url | Defines the fully qualified URL of Turing ES into which data should be indexed. | http://localhost:2700 |
| turing.site | Turing Semantic Navigation Site Name. | Sample |
| turing.weight.field | Field's name where the weight of the documents will be written. If it is empty no field will be used. | - |
| turing.auth | Whether to enable HTTP basic authentication for communicating with Turing ES. Use the `username` and `password` properties to configure your credentials. | true |
| turing.username | The username of Turing ES server. | admin |
| turing.password | The password of Turing ES server. | admin |
| turing.timestamp.field | Field used to store the timestamp of indexing. | tstamp |
| turing.field.`FIELD_NAME` | Modify or create a custom field during indexing. | - |

**Precedence of Semantic Navigation Site**

You can change the Semantic Navigation Site in the following ways:

1. Change using `solr.server.url` where is Turing URL + "/" + Turing Semantic Navigation Site, via `nutch-site.xml` or as a command line parameter. This setting is useful when using Nutch Provider in WEM where WEM uses solr.server.url to pass information about Solr to Nutch. In the case of the Turing plugin in Nutch, it reuses this configuration to know which Turing server and which site to use.
2. Change using `turing.site`, via `nutch-site.xml` or as a command line parameter. If using `turing.force.config=true` as parameter. This setting will override `solr.server.url`.
3. Adding in the `turing.xml` file, for example: `<site url="https://viglet.com" snSite="Sample"/>`. If you have this setting, it will overwrite the Semantic Navigation Site of `solr.server.url` and `turing.site`.

### 1.2.2. Nutch 1.18

This step is only for Apache Nutch 1.18. Edit the /appl/apache/nutch/conf/index-writers.xml

```
<writers xmlns="http://lucene.apache.org/nutch"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://lucene.apache.org/nutch index-writers.xsd">
 <writer id="indexer_viglet_turing_1"
        class="com.viglet.turing.nutch.indexwriter.TurNutchIndexWriter">
        <parameters>
            <param name="url" value="http://localhost:2700" />
            <param name="site" value="Sample" />
            <param name="commitSize" value="1000" />
            <param name="weight.field" value=""/>
            <param name="auth" value="true" />
            <param name="username" value="admin" />
            <param name="password" value="admin" />
        </parameters>
        <mapping>
            <copy>
                <field source="content" dest="text"/>
                <!-- <field source="title" dest="title,search"/> -->
            </copy>
            <rename>
                <field source="metatag.description" dest="description" />
                <field source="metatag.keywords" dest="keywords" />
                <field source="metatag.charset" dest="charset" />
            </rename>
            <remove>
                <field source="segment" />
                <field source="boost" />
            </remove>
        </mapping>
    </writer>
</writers>
```

#### Parameters

Modify the following parameters:

*Table 2. index-writers.xml parameters*

| Parameter | Description | Default value |
|---|---|---|
| url | Defines the fully qualified URL of Turing ES into which data should be indexed. | http://localhost:2700 |
| site | Turing Semantic Navigation Site Name. | Sample |
| weight.field | Field's name where the weight of the documents will be written. If it is empty no field will be used. | - |
| commitSize | Defines the number of documents to send to Turing ES in a single update batch. Decrease when handling very large documents to prevent Nutch from running out of memory.<br><br>**Note**: It does not explicitly trigger a server side commit. | 1000 |
| auth | Whether to enable HTTP basic authentication for communicating with Turing ES. Use the `username` and `password` properties to configure your credentials. | true |
| username | The username of Turing ES server. | admin |

| Parameter | Description | Default value |
| --- | --- | --- |
| password | The password of Turing ES server. | admin |

## 1.3. Index a Website

### 1.3.1. Nutch Command Line

There are many ways to index a website using Apache Nutch. Learn more at https://cwiki.apache.org/confluence/display/nutch/NutchTutorial.

For example, a simple way to index https://viglet.com:

1. Nutch expects some seed URLs from where to start the crawling.

```
cd /appl/apache/nutch/
mkdir urls
echo "https://viglet.com" > urls/seed.txt
```

> **TIP** You can also limit crawling to a certain hostname etc. by setting a regular expression in `/appl/apache/nutch/runtime/local/config/regex-filter.txt`

2. Index the content with Turing ES

```
# 1.12
cd /appl/apache/nutch/
bin/crawl -i urls/ crawl-output/ 5

# 1.18
cd /appl/apache/nutch/
bin/crawl -i -s urls/ crawl-output/ 5
```

or with parameter, for instance:

```
# 1.12 (Alternative 1)
cd /appl/apache/nutch/
bin/crawl -D turing.force.config=true -D turing.site="Sample" -Dturing.locale
="en_US" -i urls/ crawl-output/ 5

# 1.12 (Alternative 2)
cd /appl/apache/nutch/
bin/crawl -D solr.server.url="http://localhost:2700/Sample" -i urls/ crawl-
output/ 5

# 1.18
cd /appl/apache/nutch/
bin/crawl -D turing.site="Sample" -i -s urls/ crawl-output/ 5
```

*Table 3. crawl Parameters*

| Parameter | Example | Description |
|---|---|---|
| -D solr.server.url | -D solr.server.url="http://localhost:2700/Sample" | Turing URL + "/" + Turing Semantic Navigation Site. |
| -D turing.force.config | -D turing.force.config=true | Use turing.url and turing.site instead of solr.sever.url |
| -D turing.url | -D turing.url="localhost:2700" | Defines the fully qualified URL of Turing ES into which data should be indexed. |
| -D turing.site | -D turing.url="Sample" | Turing Semantic Navigation Site Name. |
| -D turing.auth | -D turing.auth=false | Whether to enable HTTP basic authentication for communicating with Turing ES. Use the `username` and `password` properties to configure your credentials. |

| Parameter | Example | Description |
| --- | --- | --- |
| -D turing.username | -D turing.username="admin" | The username of Turing ES server. |
| -D turing.password | -D turing.password="admin" | The password of Turing ES server. |

### 1.3.2. Nutch Provider for WEM

Web Experience Management, version 16.2 includes an example of a Page Searchable Provider using Apache Nutch, the installation and configuration is described at http://webapp.opentext.com/piroot/wcmgt/v160200/wcmgt-aci/en/html/jsframe.htm?nutch-provider-config

You can use the same Nutch Provider for InfoFusion (`com.vignette.as.server.pluggable.service.pagesearch.nutch.NutchProvider`), but using the Nutch with Turing Plugin. In Nutch Provider Configuration at WEM Configuration Console, change the variables below:

- SOLR_URL: Fill with Turing URL, for example, http://localhost:2700, instead of Solr URL;
- NUTCH_CONFIGURATION: In the XML file, put the name Turing Semantic Navigation Site in the `core` attribute, for example:

```
<?xml version="1.0" encoding="UTF-8"?>
<nutch-config
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns="http://www.vignette.com/xmlschemas/nutch-config"
        xsi:schemaLocation="http://www.vignette.com/xmlschemas/nutch-config nutch-
config.xsd">
    <default crawlId="WEM_default" core="Sample"/>
    <configuration crawlId="WEM_en" core="Sample_EN">
        <locale name="en"/>
        <locale name="en_US"/>
    </configuration>
    <configuration crawlId="WEM_es" core="Sample_ES">
        <locale name="es"/>
    </configuration>
    <configuration crawlId="WEM_de" core="Sample_DE">
        <locale name="de"/>
    </configuration>
    <configuration crawlId="WEM_fr" core="Sample_FR">
        <locale name="fr"/>
    </configuration>
    <configuration crawlId="WEM_it" core="Sample_IT">
        <locale name="it"/>
    </configuration>
</nutch-config>
```

**IMPORTANT**

If you are using the Turing ES Semantic Navigation Site's multilingual functionality, you can repeat the Site name in the `core` for each `locale` of this setting.

**TIP**

In Nutch 1.12, if there are many sites with different semantic navigation sites, use the turing-mapping.xml file to create association between the URL definitions and the semantic navigation site, for example: <site url = "https://viglet.com" snSite = "Sample" />

# Chapter 2. Database

Command line that uses the same concept as sqoop (https://sqoop.apache.org/), to create complex queries and map attributes to index based on the result.

## 2.1. Installation

Go to https://viglet.com/turing/download/ and click on "Integration > Database Connector" link to download it.

Copy the turing-jdbc.jar file to /appl/viglet/turing/jdbc

```
mkdir -p /appl/viglet/turing/jdbc
cp turing-jdbc.jar.jar /appl/viglet/turing/jdbc
```

## 2.2. Run

To run Turing JDBC Connector executable JAR file, just execute the following line:

```
$ java -jar /appl/viglet/turing/jdbc/turing-jdbc.jar <PARAMETERS>
```

### 2.2.1. Parameters

*Table 4. Turing JDBC parameters*

| Parameter | Required | Default Value | Description |
|---|---|---|---|
| --connect, -c | yes | | Specify JDBC connect string |
| --driver, -d | yes | | Manually specify JDBC driver class to use |
| --query, -q | yes | | Import the results of statement |
| --site | yes | | Specify the Semantic Navigation Site |
| --chunk, -z | no | 100 | Number of items to be sent to the queue |
| --class-name | no | | Customized Class to modified rows |
| --deindex-before -importing | no | false | Deindex before importing |
| --encoding | no | UTF-8 | Encoding Source |
| --file-content-field | no | | Field that shows Content of File |
| --file-path-field | no | | Field with File Path |

| Parameter | Required | Default Value | Description |
|---|---|---|---|
| --file-size-field | no | | Field that shows Size of File in bytes |
| --help | no | | Print usage instructions |
| --include-type-in-id, -i | no | false | Include Content Type name in Id |
| --max-content-size | no | 5 | Maximum size that content can be indexed (megabytes) |
| --multi-valued-field | no | | Multi Valued Fields |
| --password, -p | no | | Set authentication password |
| --remove-html-tags -field | no | | Remove HTML Tags into content of field |
| --server, -s | no | http://localhost:2700 | Viglet Turing Server |
| --show-output, -o | no | false | Show Output |
| --type, -t | no | CONTENT_TYPE | Set Content Type name |
| --username, -u | no | | Set authentication username |

### 2.2.2. Example

```
java -jar ./turing-jdbc.jar --deindex-before-importing true \
--include-type-in-id true -z 1 \
--file-path-field filePath --file-content-field text \
--file-size-field fileSize -t Document \
--multi-valued-separator ";" --multi-valued-field field1,field2 \
--class-name com.viglet.turing.tool.ext.TurJDBCCustomSample \
-d com.mysql.jdbc.Driver -c jdbc:mysql://localhost/sampleDB  \
```

```
-q "select * from sampleTable" -u sampleUser -p samplePassword
```

# Chapter 3. File System

Command line to index files, extracting text from files such as Word, Excel, PDF, including images, through OCR.

## 3.1. Installation

Go to https://viglet.com/turing/download/ and click on "Integration > FileSystem Connector" link to download it.

Copy the turing-filesystem.jar file to /appl/viglet/turing/fs

```
mkdir -p /appl/viglet/turing/fs
cp turing-filesystem.jar /appl/viglet/turing/fs
```

## 3.2. Run

To run Turing FileSystem Connector executable JAR file, just execute the following line:

```
$ java -jar /appl/viglet/turing/fs/turing-filesystem.jar <PARAMETERS>
```

### 3.2.1. Example

```
$ java -jar build/libs/turing-filesystem.jar --server http://localhost:2700 --nlp
b2b4a1ff-3ea3-4cec-aa95-f54d0f5f3ff8 --source-dir /appl/myfiles --output-dir
/appl/results
```

# Chapter 4. Wordpress

Wordpress plugin that allows you to index posts.

## 4.1. Installation

1. Upload the `turing4wp` folder to the `/wp-content/plugins/` directory
2. Activate the plugin through the 'Plugins' menu in WordPress
3. Configure the plugin with the hostname, port, and URI path to your Solr installation.
4. Load all your posts and/or pages via the "Load All Posts" button in the settings page.