TikaEntityProcessor

TikaEntityProcessor

⚠️ Solr3.1

Simple configuration

```
<dataConfig>
  <document>
   <dataSource type="BinURLDataSource" name="bin"/>
   <entity processor="TikaEntityProcessor" tikaConfig="tikaconfig.xml" url="${some.var.goes.here}" dataSource="
bin" format="text">
      <!--Do appropriate mapping here  meta="true" means it is a metadata field -->
      <field column="Author" meta="true" name="author"/>
      <field column="title" meta="true" name="docTitle"/>
      <!--'text' is an implicit field emited by TikaEntityProcessor . Map it appropriately-->
      <field column="text"/>
    </entity>
    <document>
</dataConfig>
```

attributes

- url : (required) The url to the source. This depends on the DataSource being used
- tikaConfig : (optional).The tika config file . If missing , default config is used. If the path is relative it is w.r.t the conf dir.
- format : (optional) output format. values are text|xml|html|none . default is 'text'. irrespective of the format, the body is emitted as a field called 'text'. Just that the content format would be different. Use 'none' if the body is not to be parsed i.e only metadata is emitted.
- parser : (optional) Default is org.apache.tika.parser.AutoDetectParser . Povide a FQN of a class which implements org.apache.tika.parser.Parser

fields

Each field may have an optional attribute meta="true". Which means this field is to be obtained from the MetaData of the document. The column value is used as the key on metadata. Checkout the list of available keys from here DublinCore , MSOffice

Finding field names

To access fields that are not part of Dublin Core or OOXML you need to find their names. One quick way is to download the commandline version of Tika here and run it against the file in question.

```
java -jar tika-app-1.5.jar ff-1923-12.docx
```

Looking at the output we see that the custom meta tag called 'Testmeta' is named 'custom:Testmeta'. Output contains UTF-8 and may look strange on the console.

```
<meta name="custom:Testmeta" content="InnehÃ¥ll"/>
```

DataSource

use any DataSource of type DataSource<InputStream>. The inbuilt ones are

- !BinURLDataSource : use for both http as well as for files
- BinContentStreamDataSource : Use for uploading content
- BinFileDataSource : use for reading from file system

Advanced Parsing

The TikaEntityProcessor can be nested with XPathEntityProcessor for indexing documents partly

example:

```
<dataConfig>
  <document>
   <dataSource type="BinURLDataSource" name="bin"/>
   <dataSource type="FieldReaderDataSource" name="fld"/>
   <entity processor="TikaEntityProcessor" tikaConfig="tikaconfig.xml" url="${some.var.goes.here}" dataSource="
bin" format="html" rootEntity="false">
      <!--Do appropriate mapping here  meta="true" means it is a metadata field -->
      <field column="Author" meta="true" name="author"/>
      <field column="title" meta="true" name="docTitle"/>
      <!--'text' is an implicit field emited by TikaEntityProcessor . Map it appropriately-->
      <field column="text"/>
      <entity type="XPathEntityProcessor" forEach="/html" dataField="text">
         <field xpath="//div"  column="foo"/>
         <field xpath="//h1"  column="h1" />
      </entity>
    </entity>
    <document>
</dataConfig>
```

Handling custom document fields

The ability to add custom metadata to OOXML documents greatly simplifies document management and searching. Assuming we want to index a custom meta field called 'Testmeta'. You need to find out its name (see above). Then add it to the data-config.xml:

```
<dataConfig>
    <dataSource type="BinFileDataSource" />
        <document>
            <entity name="files" dataSource="null" rootEntity="false"
            processor="FileListEntityProcessor"
            baseDir="/tmp/docs" fileName=".*.(doc)|(pdf)|(docx)"
            onError="skip"
            recursive="true">
                <field column="fileAbsolutePath" name="lux_uri" />
                <field column="fileSize" name="size" />
                <field column="fileLastModified" name="lastModified" />

                <entity
                    name="documentImport"
                    processor="TikaEntityProcessor"
                    url="${files.fileAbsolutePath}"
                    format="text">
                    <field column="file" name="fileName"/>
                    <field column="Author" name="author" meta="true"/>
                    <field column="title" name="title" meta="true"/>
                    <field column="text" name="text"/>
                    <field column="custom:Testmeta" name="Testmeta" meta="true"/>
                    <field column="LastModifiedBy" name="LastModifiedBy" meta="true"/>
                </entity>
        </entity>
        </document>
</dataConfig>
```

and schema.xml:

```
<fields>
 ...
<field name="Testmeta" type="text" indexed="true" stored="true" />
</fields>
```

Run DataImport with debug active Dataimport to verify that the 'Testmeta' field is seen:

```
    "Testmeta": [
      "Innehåll"
    ],
    "_version_": [
      1462987366096437200
    ]
```