gransk Documentation

Release 0.1a

Petter Chr. Bjelland

January 07, 2017

Contents

CHAPTER 1

gransk.boot

Modules for starting the gransk application.

1.1 gransk.boot.run

1.2 gransk.boot.ui

CHAPTER 2

gransk.core

2.1 gransk.core.abstract_subscriber

class `gransk.core.abstract_subscriber.`**`Subscriber`**(*pipeline*)

Bases: `object`

Abstract class subscribers inherits from.

Add subscriber to pipeline.

Parameters **`pipeline`** (`gransk.core.pipeline.Pipeline`) – Pipeline managing subscribers and events.

`CONSUMES` **= None**

Subscribe to the following list of topics [`unicode`]

`MAGIC` **= None**

Documents starting whith these bytes should be passed to this subscriber.

`SERVICE_ID` **= None**

This subscriber may be fetched from the pipeline by this ID.

`consume`(*doc*, *payload*)

Abstract method for receiving data.

Parameters

- **`doc`** (`gransk.core.document.Document`) – The document the event belongs to.
- **`payload`** (`file`) – File pointer beloning to the document.

`produce`(*topic*, *doc*, *payload*)

Add a new event to the pipeline.

Parameters

- **`topic`** (`unicode`) – Topic to add event to.
- **`doc`** (`gransk.core.document.Document`) – The document the event belongs to.
- **`payload`** (`file`) – File pointer beloning to the document.

`setup`(*config*)

Placeholder for configuration of subscriber, before receiving data.

Parameters **`config`** (`dict`) – Configuration object for processing.

stop()

Stop subscriber after progressing is completed.

2.2 gransk.core.bootstrap

2.3 gransk.core.detect_type

class gransk.core.detect_type.**Subscriber**(*pipeline*)

Bases: *gransk.core.abstract_subscriber.Subscriber*

Class for determining document type.

Add subscriber to pipeline.

Parameters **pipeline** (gransk.core.pipeline.Pipeline) – Pipeline managing subscribers and events.

consume(*doc*, *payload*)

Determine document type, either by extension or based on Tika mimetype. Produces an event based on the found type.

Parameters

- **doc** (gransk.core.document.Document) – Document to process.
- **payload** – File pointer to the document.

setup(*config*)

Generate file extension-based type detection from the given configuration.

Parameters **config** (dict) – Configuration object.

2.4 gransk.core.document

2.5 gransk.core.file_collector

class gransk.core.file_collector.**Collector**(*config*)

Bases: object

Class for collecting paths from filesystem.

collect(*root_path*)

Collect all files matching a path recursively.

Parameters **root_path** – Input path. May point to a file or directory.

Returns Iterator of found paths.

2.6 gransk.core.helper

Module containing string variables used throughout the processing.

2.7 gransk.core.injector

2.8 gransk.core.magic

class `gransk.core.magic.`**`Subscriber`**(*pipeline*)

Bases: *`gransk.core.abstract_subscriber.Subscriber`*

Identify extractor subscribers based on file header.

Add subscriber to pipeline.

Parameters **`pipeline`** (`gransk.core.pipeline.Pipeline`) – Pipeline managing subscribers and events.

`consume`(*doc*, *payload*)

Identify extractors and call their callback functions.

Parameters

- **`doc`** (`gransk.core.document.Document`) – The document object.
- **`payload`** (`file`) – The document object.

`setup`(_)

Compile file headers for all magic extractors into a regex pattern.

2.9 gransk.core.pipeline

class `gransk.core.pipeline.`**`Pipeline`**

Bases: `object`

Class for instatiating and managing subscribers and events during processing. Subscribers are registered to topics (`str`). A subscriber may register to any number of topics, or a magic header. When a subscriber produces an event, the pipeline finds all subscribers for that event topic and calls these (their `consume(doc, payload)` function) one by one. See `gransk.core.abstract_subscriber.Subscriber.CONSUME`.

During text extraction, we may want to implement custom extractors. This is done by registering to a magic header, which means the first N bytes of the document. See `gransk.core.abstract_subscriber.Subscriber.MAGIC`.

`get_service`(*service_id*)

Get service by ID.

Parameters **`service_id`** (`str`) – ID of service to fetch.

Returns `object` service. None if no service is found.

`produce`(*topic*, *doc*, *payload*)

Produce a new event.

Parameters

- **`topic`** (`str`) – The topic of the produced event.
- **`doc`** (`gransk.core.Document`) – The document to which the event belongs.
- **`payload`** (`file`) – The file pointer beloning to the document.

`register_listener`(*topic*, *callback*)

Register a subscriber callback to a topic.

Parameters

- **`topic`** (`str`) – The topic to subscribe to.
- **`callback`** (`function`) – Function to call when an event with this topic is produced.

`register_magic`(*magic*, *subscriber*)

Register a subscriber to a magic header.

Parameters

- **`magic`** – The header of files to subscribe to.
- **`subscriber`** – The subscriber object.

`register_service`(*service_id*, *service*)

Register a subscriber as a service that is fetchable by ID. There may only be a single service with a given ID.

Parameters

- **`service_id`** (`str`) – The ID of the service.
- **`service`** (`object`) – The service object.

`stop`()

Stop all subscribers.

`granskcore.pipeline.`**`build_pipeline`**(*config*)

Build the pipeline based on the given configuration.

Parameters **`config`** (`dict`) – The configuration object.

Returns Instantiated `gransk.core.pipeline.Pipeline`

`gransk.core.pipeline.`**`init_subscriber`**(*config*, *subscriber_mod*, *pipeline*)

Instatiate a Subscriber object and add it to the pipeline.

Parameters

- **`subscriber_mod`** (`str`) – Reference to the module containing the Subscriber.
- **`pipline`** (`gransk.core.Pipeline`) – The pipeline object to add the subscriber to.

2.10 gransk.core.process

class `gransk.core.process.`**`Subscriber`**(*pipeline*)

Bases: *`gransk.core.abstract_subscriber.Subscriber`*

Module for producing common processing events on a document.

Add subscriber to pipeline.

Parameters **`pipeline`** (`gransk.core.pipeline.Pipeline`) – Pipeline managing subscribers and events.

`consume`(*doc*, *_*)

Run a document through processing events.

Parameters **`doc`** (`gransk.core.document.Document`) – Document to process.

gransk.plugins

3.1 gransk.plugins.analysis

Compute useful things from stuff.

3.1.1 gransk.plugins.analysis.abstract_related

class `gransk.plugins.analysis.abstract_related.`**`Subscriber`**`(`*pipeline*`)`

Bases: *`gransk.core.abstract_subscriber.Subscriber`*

Find entities and documents that are related to each other based on found entities.

Add subscriber to pipeline.

Parameters **`pipeline`** (`gransk.core.pipeline.Pipeline`) – Pipeline managing subscribers and events.

`consume``(`*doc*, *payload*`)`

Abstract method that when implemented should add data from documents.

Parameters

- **`doc`** (`gransk.core.document.Document`) – Document object.
- **`payload`** (`file`) – File pointer beloning to document.

`get_related_to``(`*_id*, *min_score=None*, *min_shared=None*, *max_results=None*`)`

Get objects related to the given ID, based on:

- How many entities they have in common (related documents)
- How many documents they have in common (related entities)

Parameters

- **`_id`** (`str`) – ID of the object (entity or document) to get related for.
- **`min_score`** (`float`) – shared / min(a_frequency, b_frequency).
- **`min_shared`** (`int`) – Minimum amount of shared documetns or entities.
- **`max_results`** (`int`) – Maximum number of related objects to return.

`load_all``(`*config*`)`

Load all existing data.

Parameters **config** (dict) – Configuration object.

setup (*config*)
Load existing data for given worker.

Parameters **config** (dict) – Configuration object.

stop ()
Write data to file.

3.1.2 gransk.plugins.analysis.entity_network

class gransk.plugins.analysis.entity_network.**Subscriber** (*pipeline*)
Bases: *gransk.core.abstract_subscriber.Subscriber*

Class computing network surrounding an entity.

Add subscriber to pipeline.

Parameters **pipeline** (gransk.core.pipeline.Pipeline) – Pipeline managing subscribers and events.

consume (*doc*, *payload*)
Ignored.

get_for (*entity_id*, *hops=1*)
Get network around the given entity ID.

Parameters

- **entity_id** (str) – Entity to get network for.
- **hops** (int) – Maximum distance of included nodes from the given entity.

Returns dict

setup (*config*)
Loads services for related entities and documents.

Parameters **config** (dict) – Configuration object.

3.1.3 gransk.plugins.analysis.related_documents

class gransk.plugins.analysis.related_documents.**Subscriber** (*pipeline*)
Bases: *gransk.plugins.analysis.abstract_related.Subscriber*

Class for finding related documents based on the entities they have in common.

Add subscriber to pipeline.

Parameters **pipeline** (gransk.core.pipeline.Pipeline) – Pipeline managing subscribers and events.

consume (*doc*, *_*)
Add all entities to the reference set.

Parameters **doc** (gransk.core.document.Document) – Document object.

3.1.4 gransk.plugins.analysis.related_entities

class `gransk.plugins.analysis.related_entities.`**`Subscriber`**(*pipeline*)

Bases: *`gransk.plugins.analysis.abstract_related.Subscriber`*

Class for finding related entities based on the documents they have in common.

Add subscriber to pipeline.

Parameters **`pipeline`** (`gransk.core.pipeline.Pipeline`) – Pipeline managing subscribers and events.

`consume`(*doc*, *_*)

Add document to each entity's reference set.

Parameters **`doc`** (`gransk.core.document.Document`) – Document object.

3.2 gransk.plugins.extractors

Extract content from document.

3.2.1 gransk.plugins.extractors.ewf_strings

3.2.2 gransk.plugins.extractors.file_meta

class `gransk.plugins.extractors.file_meta.`**`Subscriber`**(*pipeline*)

Bases: *`gransk.core.abstract_subscriber.Subscriber`*

Class for extracting metadata from documents using Apache Tika.

Add subscriber to pipeline.

Parameters **`pipeline`** (`gransk.core.pipeline.Pipeline`) – Pipeline managing subscribers and events.

`consume`(*doc*, *payload*)

Upload document to Apache Tika and parse results.

Parameters

- **`doc`** (`gransk.core.document.Document`) – Document object.
- **`payload`** (`file`) – File pointer beloning to document.

`setup`(*config*)

Load mediatype mapping from file. This is used to determine document type.

Parameters **`config`** (`dict`) – Configuration object.

3.2.3 gransk.plugins.extractors.picture_meta

class `gransk.plugins.extractors.picture_meta.`**`Subscriber`**(*pipeline*)

Bases: *`gransk.core.abstract_subscriber.Subscriber`*

Determine width and height. Called when the document is a picture.

Add subscriber to pipeline.

Parameters **pipeline** (`gransk.core.pipeline.Pipeline`) – Pipeline managing subscribers and events.

consume (*doc, payload*)

Parse picture header and extract width/height information.

Parameters

- **doc** (`gransk.core.document.Document`) – Document object.
- **payload** (`file`) – File pointer beloning to document.

setup (*config*)

Define picture magic headers and compute regex pattern to find correct parser later.

Parameters **config** (`dict`) – Configuration object.

3.2.4 gransk.plugins.extractors.strings

3.2.5 gransk.plugins.extractors.tika_extractor

class `gransk.plugins.extractors.tika_extractor.`**Subscriber** (*pipeline*)

Bases: *gransk.core.abstract_subscriber.Subscriber*

Class for uploading documents to Apache Tika and reading text response. Tika is an open source tool that is capable of parsing a vast number (>200) of document formats.

Add subscriber to pipeline.

Parameters **pipeline** (`gransk.core.pipeline.Pipeline`) – Pipeline managing subscribers and events.

consume (*doc, payload*)

Upload document to Apache Tika and add result to document as text.

Parameters

- **doc** (`gransk.core.document.Document`) – Document object.
- **payload** (`file`) – File pointer beloning to document.

setup (*config*)

Define maximum size of document to upload.

Parameters **config** (`dict`) – Configuration object.

3.3 gransk.plugins.find

Find stuff in extracted content.

3.3.1 gransk.plugins.find.find_entities

class `gransk.plugins.find.find_entities.`**Subscriber** (*pipeline*)

Bases: *gransk.core.abstract_subscriber.Subscriber*

Class for finding entities in text based on regular expressions.

Add subscriber to pipeline.

Parameters **pipeline** (gransk.core.pipeline.Pipeline) – Pipeline managing subscribers and events.

consume (*doc*, *_*)
Find entities in documents matching compiled regular expression.

Parameters **doc** (gransk.core.document.Document) – Document object.

setup (*config*)
Compile configured regular expressions.

Parameters **config** (dict) – Configuration object.

3.3.2 gransk.plugins.find.find_names_brute

class gransk.plugins.find.find_names_brute.**Subscriber** (*pipeline*)
Bases: *gransk.core.abstract_subscriber.Subscriber*

Class for finding names in text based on a provided list of tokens. This approach has the benefit over other Named Entity Extraction approaches that it is independent of the context in which the names are. It may thus be a good supplement to improve entity recognition.

Add subscriber to pipeline.

Parameters **pipeline** (gransk.core.pipeline.Pipeline) – Pipeline managing subscribers and events.

consume (*doc*, *_*)
Find names in documents based on the provided word list.

Parameters **doc** (gransk.core.document.Document) – Document object.

setup (*config*)
Load name model (word list) and compile regexes for stop characters.

Parameters **config** (dict) – Configuration object.

3.3.3 gransk.plugins.find.polyglot_ner

class gransk.plugins.find.polyglot_ner.**Subscriber** (*pipeline*)
Bases: *gransk.core.abstract_subscriber.Subscriber*

Class for finding named entities in text using the Polyglot NER package.

Add subscriber to pipeline.

Parameters **pipeline** (gransk.core.pipeline.Pipeline) – Pipeline managing subscribers and events.

consume (*doc*, *_*)
Find names in documents using Polyglot NER.

Parameters **doc** (gransk.core.document.Document) – Document object.

setup (*config*)
Load Polyglot NER pakcage.

Parameters **config** (dict) – Configuration object.

3.4 gransk.plugins.storage

Store stuff places.

3.4.1 gransk.plugins.storage.copy_file

3.4.2 gransk.plugins.storage.copy_picture

3.4.3 gransk.plugins.storage.es_index

class `gransk.plugins.storage.es_index.`**`Subscriber`**(*pipeline*)

Bases: *`gransk.core.abstract_subscriber.Subscriber`*

Class for adding documents to an Elasticsearch cluster.

Add subscriber to pipeline.

Parameters **`pipeline`** (`gransk.core.pipeline.Pipeline`) – Pipeline managing subscribers and events.

`consume`(*doc*, *_*)

Add document to Elasticsearch.

Parameters **`doc`** (`gransk.core.document.Document`) – Document object.

`create_mapping`()

Create index mappig in Elasticsearch cluster.

`setup`(*config*)

Establish connection to Elasticsearch cluster and start periodic commit.

Parameters **`config`** (`dict`) – Configuration object.

`stop`()

Commit current remaning documents.

3.4.4 gransk.plugins.storage.store_text

3.5 gransk.plugins.unpackers

Unpack containers (disk images and archives).

3.5.1 gransk.plugins.unpackers.diskimage_reader

3.5.2 gransk.plugins.unpackers.unpack_archive

3.5.3 gransk.plugins.unpackers.unpack_diskimage

Python Module Index

Index