mf2util Documentation

Release 0.1.0

Kyle Mahan

Sep 27, 2017

Contents

Utilities for interpreting mf2 data.

Microformats2 is a general way to mark up any HTML document with classes and propeties. This module uses domain-specific assumptions about the classes (specifically h-entry and h-event) to extract certain interesting properties.

class mf2util.**FixedOffset**(*offset*, *name*)
A class building tzinfo objects for fixed-offset time zones. Note that FixedOffset(0, "UTC") is a different way to build a UTC tzinfo object.

Fixed offset in minutes east from UTC. from Python 2 documentation https://docs.python.org/2/library/datetime.html#tzinfo-objects

class mf2util.**UTC**
UTC timezone, from Python documentation https://docs.python.org/2/library/datetime.html#tzinfo-objects

mf2util.**classify_comment**(*parsed*, *target_urls*)
Find and categorize comments that reference any of a collection of target URLs. Looks for references of type reply, like, and repost.

Parameters

- **parsed** (*dict*) – a mf2py parsed dict
- **target_urls** (*list*) – a collection of urls that represent the target post. this can include alternate or shortened URLs.

Returns a list of applicable comment types ['like', 'reply', 'repost']

mf2util.**convert_relative_paths_to_absolute**(*source_url*, *base_href*, *html*)
Attempt to convert relative paths in foreign content to absolute based on the source url of the document. Useful for displaying images or links in reply contexts and comments.

Gets list of tags/attributes from *URL_ATTRIBUTES*. Note that this function uses a regular expression to avoid adding a library dependency on a proper parser.

Parameters

- **source_url** (*str*) – the source of the parsed document.
- **html** (*str*) – the text of the source document

Returns the document with relative urls replaced with absolute ones

mf2util.**find_all_entries**(*parsed*, *types*, *include_properties=False*)
Find all h-* objects of a given type in BFS-order. Traverses the top-level items and their children and descendents. Includes property values (e.g. finding all h-cards would not find values of "p-author h-card") only if *include_properties* is True.

Parameters

- **parsed** (*dict*) – a mf2py parsed dict
- **types** (*list*) – target types, e.g. ['h-entry', 'h-event']
- **include_properties** (*boolean*) – include properties in search of entries

Returns all entries with any of the the target types

mf2util.**find_author**(*parsed*, *source_url=None*, *hentry=None*, *fetch_mf2_func=None*)
Use the authorship discovery algorithm https://indiewebcamp.com/authorship to determine an h-entry's author.

Parameters

- **parsed** (*dict*) – an mf2py parsed dict.

- **source_url** (*str*) – the source of the parsed document.
- **dict** (*hentry*) – optional, the h-entry we're examining, if omitted, we'll just use the first one
- **callable** (*fetch_mf2_func*) – optional function that takes a URL and returns parsed mf2

Returns a dict containing the author's name, photo, and url

mf2util.**find_datetimes**(*parsed*)
Find published, updated, start, and end dates.

Parameters **parsed** (*dict*) – a mf2py parsed dict

Returns a dictionary from property type to datetime or date

mf2util.**find_first_entry**(*parsed*, *types*)
Find the first interesting h-* object in BFS-order

Parameters

- **parsed** (*dict*) – a mf2py parsed dict
- **types** (*list*) – target types, e.g. ['h-entry', 'h-event']

Returns an mf2py item that is one of *types*, or None

mf2util.**get_plain_text**(*values*, *strip=True*)
Get the first value in a list of values that we expect to be plain-text. If it is a dict, then return the value of "value".

Parameters

- **values** (*list*) – a list of values
- **strip** (*boolean*) – true if we should strip the plaintext value

Returns a string or None

mf2util.**interpret**(*parsed*, *source_url*, *base_href=None*, *item=None*, *use_rel_syndication=True*, *want_json=False*, *fetch_mf2_func=None*)
Interpret a permalink of unknown type. Finds the first interesting h-* element, and delegates to *interpret_entry()* if it is an h-entry or *interpret_event()* for an h-event

Parameters

- **parsed** (*dict*) – the result of parsing a mf2 document
- **source_url** (*str*) – the URL of the source document (used for authorship discovery)
- **base_href** (*str*) – (optional) the href value of the base tag
- **item** (*dict*) – (optional) the item to be parsed. If provided, this will be used instead of the first element on the page.
- **use_rel_syndication** (*boolean*) – (optional, default True) Whether to include rel=syndication in the list of syndication sources. Sometimes useful to set this to False when parsing h-feeds that erroneously include rel=syndication on each entry.
- **want_json** (*boolean*) – (optional, default False) If true, the result will be pure json with datetimes as strings instead of python objects
- **fetch_mf2_func** (*callable*) – (optional) function to fetch mf2 parsed output for a given URL.

Returns a dict as described by interpret_entry or interpret_event, or None

mf2util.**interpret_comment**(*parsed*, *source_url*, *target_urls*, *base_href=None*, *want_json=False*, *fetch_mf2_func=None*)

Interpret received webmentions, and classify as like, reply, or repost (or a combination thereof). Returns a dict as described in *interpret_entry()*, with the additional fields:

```
{
 'comment_type': a list of strings, zero or more of
                 'like', 'reply', or 'repost'
 'rsvp': a string containing the rsvp response (optional)
}
```

Parameters

- **parsed** (*dict*) – a parsed mf2 parsed document
- **source_url** (*str*) – the URL of the source document
- **target_urls** (*list*) – a collection containing the URL of the target document, and any alternate URLs (e.g., shortened links) that should be considered equivalent when looking for references
- **base_href** (*str*) – (optional) the href value of the base tag
- **want_json** (*boolean*) – (optional, default False) If true, the result will be pure json with datetimes as strings instead of python objects
- **fetch_mf2_func** (*callable*) – (optional) function to fetch mf2 parsed output for a given URL.

Returns a dict as described above, or None

mf2util.**interpret_entry**(*parsed*, *source_url*, *base_href=None*, *hentry=None*, *use_rel_syndication=True*, *want_json=False*, *fetch_mf2_func=None*)

Given a document containing an h-entry, return a dictionary:

```
{
 'type': 'entry',
 'url': the permalink url of the document (may be different than source_url),
 'published': datetime or date,
 'updated': datetime or date,
 'name': title of the entry,
 'content': body of entry (contains HTML),
 'author': {
  'name': author name,
  'url': author url,
  'photo': author photo
 },
 'syndication': [
   'syndication url',
   ...
 ],
 'in-reply-to': [...],
 'like-of': [...],
 'repost-of': [...],
}
```

Parameters

- **parsed** (*dict*) – the result of parsing a document containing mf2 markup

- **source_url** (*str*) – the URL of the parsed document, used by the authorship algorithm
- **base_href** (*str*) – (optional) the href value of the base tag
- **hentry** (*dict*) – (optional) the item in the above document representing the h-entry. if provided, we can avoid a redundant call to find_first_entry
- **use_rel_syndication** (*boolean*) – (optional, default True) Whether to include rel=syndication in the list of syndication sources. Sometimes useful to set this to False when parsing h-feeds that erroneously include rel=syndication on each entry.
- **want_json** (*boolean*) – (optional, default False) if true, the result will be pure json with datetimes as strings instead of python objects
- **fetch_mf2_func** (*callable*) – (optional) function to fetch mf2 parsed output for a given URL.

Returns a dict with some or all of the described properties

mf2util.**interpret_event**(*parsed*, *source_url*, *base_href=None*, *hevent=None*, *use_rel_syndication=True*, *want_json=False*, *fetch_mf2_func=None*)

Given a document containing an h-event, return a dictionary:

```
{
 'type': 'event',
 'url': the permalink url of the document (may be different than source_url),
 'start': datetime or date,
 'end': datetime or date,
 'name': plain-text event name,
 'content': body of event description (contains HTML)
}
```

Parameters

- **parsed** (*dict*) – the result of parsing a document containing mf2 markup
- **source_url** (*str*) – the URL of the parsed document, not currently used
- **base_href** (*str*) – (optional) the href value of the base tag
- **hevent** (*dict*) – (optional) the item in the above document representing the h-event. if provided, we can avoid a redundant call to find_first_entry
- **use_rel_syndication** (*boolean*) – (optional, default True) Whether to include rel=syndication in the list of syndication sources. Sometimes useful to set this to False when parsing h-feeds that erroneously include rel=syndication on each entry.
- **want_json** (*boolean*) – (optional, default false) if true, the result will be pure json with datetimes as strings instead of python objects
- **fetch_mf2_func** (*callable*) – (optional) function to fetch mf2 parsed output for a given URL.

Returns a dict with some or all of the described properties

mf2util.**interpret_feed**(*parsed*, *source_url*, *base_href=None*, *hfeed=None*, *want_json=False*, *fetch_mf2_func=None*)

Interpret a source page as an h-feed or as an top-level collection of h-entries.

Parameters

- **parsed** (*dict*) – the result of parsing a mf2 document

- **source_url** (*str*) – the URL of the source document (used for authorship discovery)
- **base_href** (*str*) – (optional) the href value of the base tag
- **hfedd** (*dict*) – (optional) the h-feed to be parsed. If provided, this will be used instead of the first h-feed on the page.
- **fetch_mf2_func** (*callable*) – (optional) function to fetch mf2 parsed output for a given URL.

Returns a dict containing 'entries', a list of entries, and possibly other feed properties (like 'name').

mf2util.**is_name_a_title**(*name*, *content*)

Determine whether the name property represents an explicit title.

Typically when parsing an h-entry, we check whether p-name == e-content (value). If they are non-equal, then p-name likely represents a title.

However, occasionally we come across an h-entry that does not provide an explicit p-name. In this case, the name is automatically generated by converting the entire h-entry content to plain text. This definitely does not represent a title, and looks very bad when displayed as such.

To handle this case, we broaden the equality check to see if content is a subset of name. We also strip out non-alphanumeric characters just to make the check a little more forgiving.

Parameters

- **name** (*str*) – the p-name property that may represent a title
- **content** (*str*) – the plain-text version of an e-content property

Returns True if the name likely represents a separate, explicit title

mf2util.**parse_author**(*obj*)

Parse the value of a u-author property, can either be a compound h-card or a single name or url.

Parameters **obj** (*object*) – the mf2 property value, either a dict or a string

Result a dict containing the author's name, photo, and url

mf2util.**parse_datetime**(*s*)

The definition for microformats2 dt-* properties are fairly lenient. This method converts an mf2 date string into either a datetime.date or datetime.datetime object. Datetimes will be naive unless a timezone is specified.

Parameters **s** (*str*) – a mf2 string representation of a date or datetime

Returns datetime.date or datetime.datetime

Raises **ValueError** – if the string is not recognizable

mf2util.**parse_dt**(*s*)

The definition for microformats2 dt-* properties are fairly lenient. This method converts an mf2 date string into either a datetime.date or datetime.datetime object. Datetimes will be naive unless a timezone is specified.

Parameters **s** (*str*) – a mf2 string representation of a date or datetime

Returns datetime.date or datetime.datetime

Raises **ValueError** – if the string is not recognizable

mf2util.**post_type_discovery**(*hentry*)

Implementation of the post-type discovery algorithm defined here https://indiewebcamp.com/post-type-discovery#Algorithm

Parameters **hentry** (*dict*) – mf2 item representing the entry to test

Returns string, one of: 'org', 'person', 'event', 'rsvp', 'invite', 'reply', 'repost', 'like', 'photo', 'article', note'

mf2util.**representative_hcard**(*parsed*, *source_url*)
Find the representative h-card for a URL

http://microformats.org/wiki/representative-h-card-parsing

Parameters

- **parsed** (*dict*) – an mf2 parsed dict
- **source_url** (*str*) – the source of the parsed document.

Returns the representative h-card if one is found

mf2util.**timezone_from_offset**
alias of *FixedOffset*

Python Module Index

m

Index