
PyPresseportal

Release 0.0.1

Timo Cornelius Metzger

Aug 05, 2020

CONTENTS:

- 1 A Python interface into the presseportal.de API** **1**

- 2 Scope of this document** **3**
 - 2.1 Installing PyPresseportal 3
 - 2.2 Getting started with PyPresseportal 3
 - 2.3 Usage examples 5
 - 2.4 Modules Documentation 9

- 3 Indices and tables** **19**

- Python Module Index** **21**

- Index** **23**

A PYTHON INTERFACE INTO THE PRESSEPORTAL.DE API

PyPresseportal is a Python wrapper for the API of the website presseportal.de. With PyPresseportal, you can use Python objects to easily access the website's extensive data resources.

The website presseportal.de is a service provided by 'news aktuell', owned by dpa (Deutsche Presse Agentur). It is one of the biggest distributors of press releases and investor relations announcements in Germany. For example, almost all police and fire departments use this service to distribute their press releases.

PyPresseportal is in no way connected to presseportal.de, 'news aktuell' or dpa. PyPresseportal is independently developed by volunteers as an Open Source library.

Note: A valid API key from presseportal.de is required to access data. You can find information on the API's Terms of use and apply for an API key at <https://api.presseportal.de/en>.

SCOPE OF THIS DOCUMENT

This documentation describes how to use PyPresseportal to access data from presseportal.de in Python. You will learn how to install PyPresseportal, initialize the API with the `pypresseportal.PresseportalApi` class, and use PyPresseportal's various methods to search, request, and parse data from the presseportal.de API.

2.1 Installing PyPresseportal

2.1.1 Installing with `pip`

Use `pip` on a command line to download PyPresseportal from PyPI and install it on your system:

```
$ pip install pypresseportal
```

2.1.2 Downloading from GitHub

You can also manually download the most recent version of PyPresseportal from [GitHub](https://github.com/tcmetzger/pypresseportal) using `git clone`:

```
$ git clone https://github.com/tcmetzger/pypresseportal.git
$ python setup.py install
```

2.2 Getting started with PyPresseportal

Follow these steps to set up PyPresseportal and retrieve data from presseportal.de:

2.2.1 1. Getting an API key

PyPresseportal requires a valid API key from presseportal.de. You can request an API key on this website: <https://api.presseportal.de/en>. This website is also where you can find general information on the API, as well as the API's Terms.

2.2.2 2. Initializing the API

First, create an instance of the `pypresseportal.PresseportalApi` class, using your API key:

```
>>> from pypresseportal import PresseportalApi
>>> api_object = PresseportalApi(YOUR_API_KEY)
```

2.2.3 3. Requesting data from the API

Next, request data from the API through the `PresseportalApi` class. This class offers several methods to access API data. The *Modules Documentation* explains all available methods in detail.

The easiest way to access the most recently published stories is the `get_stories()` method. If you do not provide any arguments to this method, PyPresseportal defaults to retrieving the 50 most recent stories available.

For example:

```
>>> from pypresseportal import PresseportalApi
>>> api_object = PresseportalApi(YOUR_API_KEY)
>>> stories = api_object.get_stories()
```

2.2.4 4. Accessing the data

`get_stories()` returns a list of `Story` objects. Access `Story` data through the individual `Story` object's attributes.

For example:

```
>>> stories[0].title
"Kohls Wohnhaus hat keinen Denkmalwert"
>>> stories[0].id
"4622388"
```

For each story the API returns, the method `get_stories()` will generate a `Story` object with the following attributes:

- `data` - The raw json data returned by the API.
- `id` - The story's unique id as generated by the API.
- `url` - URL of the story on the presseportal.de website.
- `title` - Headline of the story.
- `body` - Full text body.
- optional: `teaser` - Teaser text. Only included if `teaser=True` is passed to `get_stories()`. If a teaser is requested, it will replace the `body` attribute.
- `published` - Publication date (as datetime object).
- `language` - Language of the story (usually 'de' or 'en').
- `ressort` - Editorial department.
- `company_id` - Unique id of the company publishing the story.
- `company_url` - URL of the company publishing the story.
- `company_name` - Name of the company publishing the story.
- `keywords` - List of keywords assigned to the story.

- optional: `media` - Information on media attachments, if present.
- `highlight` - Promoted story flag (on/off).
- `short` - Shortened URL.

2.2.5 5. Learning more about PyPresseportal

The presseportal.de API offers several ways to request data. See the *Usage examples* to see PyPresseportal in action. Detailed information about PyPresseportal is available in the *Modules Documentation*.

These are the methods supported by PyPresseportal:

- `pypresseportal.PresseportalApi.get_company_information()`: Return detailed info about a specific company (requires company id).
- `pypresseportal.PresseportalApi.get_entity_search_results()`: Search for company or public service office by location or name (provides company/office id).
- `pypresseportal.PresseportalApi.get_investor_relations_news()`: Return investor relations news (Ad Hoc news, Directors' Dealings, reports, etc).
- `pypresseportal.PresseportalApi.get_investor_relations_news_company()`: Return investor relations news about a specific company (requires company id).
- `pypresseportal.PresseportalApi.get_public_service_news()`: Return stories released by public service offices (police and fire departments, etc).
- `pypresseportal.PresseportalApi.get_public_service_office_information()`: Return detailed info about a specific public service office (requires office id).
- `pypresseportal.PresseportalApi.get_public_service_specific_office()`: Return stories released by a specific public service office (requires office id).
- `pypresseportal.PresseportalApi.get_public_service_specific_region()`: Return Stories released by public service offices in a specific geographic region (list of available regions).
- `pypresseportal.PresseportalApi.get_stories_specific_company()`: Return stories released by a specific company (requires company id).
- `pypresseportal.PresseportalApi.get_stories_keywords()`: Return stories assigned to specific keywords (list of available keywords).
- `pypresseportal.PresseportalApi.get_stories_topic()`: Return stories assigned to a specific topic (list of available topics).
- `pypresseportal.PresseportalApi.get_stories()`: Return recently published stories.

2.3 Usage examples

The following examples give you an idea of what can be done with PyPresseportal, featuring some of the functionalities of the library. However, PyPresseportal has several additional functions that are not part of these examples. You can find more information about all available classes, methods, and attributes in the *Modules Documentation*.

2.3.1 Police and fire department press releases from a specific state

Use the `get_public_service_specific_region()` method to request press releases from public service offices in a specific region.

First, this method requires a `region_code` argument, as defined in this list: <https://api.presseportal.de/en/doc/value/region>. For example, "by" is the region code for Bavaria:

```
>>> from pypresseportal import PresseportalApi
>>> api_object = PresseportalApi(YOUR_API_KEY)
>>> bavaria_stories = api_object.get_public_service_specific_region(region_code="by",
↳ limit=20)
```

Note that this example also includes the optional argument `limit=20`, which limits the response to the 20 most recently published stories:

```
>>> len(bavaria_stories)
20
```

Iterate over the list to extract some of the attributes of each Story object:

```
>>> for story in bavaria_stories:
>>>     print(story.title)
>>>     print(story.body)
POL-MFR: (797) Warnung vor "falschen Polizeibeamten"
Nürnberg (ots)
In den letzten Tagen registrierte die mittelfränkische Polizei zahlreiche Hinweise
(...)
```

The *Modules Documentation* contains more information about all possible attributes a Story object can have: [pypresseportal.Story](#)

2.3.2 Press releases with images

Use the `media` attribute available to most methods in PyPresseportal to limit the API's response to stories containing a specific media type (available media types are listed at <https://api.presseportal.de/en/doc/value/media>).

This code requests the 10 most recent news stories featuring an image, by passing `media="image"`:

```
>>> from pypresseportal import PresseportalApi
>>> api_object = PresseportalApi(YOUR_API_KEY)
>>> stories_with_image = api_object.get_stories(media="image", limit=10)
```

Read the relevant attributes for each story, such as `.image` which contains json data about image media files that are part of the story:

```
>>> for story in stories_with_image:
>>>     print(story.title)
>>>     print(story.image)
Das Erste: Richtiges Rezept für den Sieg? Tim Mälzer und Steffen Henssler treten an_
↳ gegen Jörg Pilawas "Quizduell-Olymp" am Freitag, 19. Juni 2020, 18:50 Uhr im Ersten_
↳ (FOTO)
[{'id': '679851', 'url': 'https://cache.pressmailing.net/thumbnail/story_big/ce365907-
↳ da06-4542-a8ca-3fc42db21e2b/1_F313_Quizduell_Olymp_2020.jpg', 'name': '1-f313-
↳ quizduell-olymp-2020.jpg', 'size': '2371804', 'mime': 'image/jpeg', 'type': 'image',
↳ 'caption': 'ARD QUIZDUELL-OLYMP, FOLGE 313, "Steffen Henssler und Tim Mälzer", am_
↳ Freitag (19.06.20) um 18:50 Uhr im ERSTEN.\nDie Kandidaten des Teams "Köche":_
↳ Steffen Henssler (l.) und Tim Mälzer (r.), beide TV-Köche.\n© ARD/Uwe K...
↳ honorarfrei - Verwendung gemäß der AGB im engen inhaltlichen, redaktionellen_
↳ Zusammenhang mit genannter ARD-Sendung bei Nennung "Bild: ARD/Uwe Ernst" (S2). ARD_
↳ Programmredaktion/Bildredaktion, Tel: 089/5900-23534, Chapter 2. Scope of this document
↳ Weiterer Text über ots und www.presseportal.de/nr/6694 / Die Verwendung dieses_
↳ Bildes ist für redaktionelle Zwecke honorarfrei. Veröffentlichung bitte unter_
↳ Quellenangabe: "obs/ARD Das Erste"']}]
```

(continued from previous page)

2.3.3 Press releases from police and fire departments in a specific city

Combine the methods `get_entity_search_results()` and `get_public_service_specific_office()` to access all available public service office press releases for a specific city.

First, initialize the API and search for all available offices in a city. Use the argument `entity='office'` to search for public service offices. Use the argument `search_term` to pass the city you are searching for, for example “Dortmund”:

```
>>> from pypresseportal import PresseportalApi
>>> api_object = PresseportalApi(YOUR_API_KEY)
>>> search_results = api_object.get_entity_search_results(search_term=["Dortmund"],
↳entity="office")
```

Three public service offices are available in Dortmund:

```
>>> for office in search_results:
>>>     print(office.id, office.name)
4971 Polizei Dortmund
115869 Feuerwehr Dortmund
121242 Hauptzollamt Dortmund
```

Next, store the offices’ `.id` attributes in a list:

```
>>> local_office_ids = []
>>> for data in search_results:
>>>     local_office_ids.append(data.id)
```

Finally, use `get_public_service_specific_office()` to access the most recent press releases of those three offices:

```
>>> for office in local_office_ids:
>>>     office_data = api_object.get_public_service_specific_office(id=office,
↳limit=4)
>>>     for story in office_data:
>>>         print(story.title)
>>>         print(story.body)
POL-DO: Mehrere Fahrzeuge in Dortmund-Dorstfeld beschädigt - Polizei sucht Zeugen
Dortmund (ots) - Lfd. Nr.: 0628
(...)
```

Instead of `get_public_service_specific_office()`, you can use `get_stories_specific_company()` to access press releases of a specific company, or `get_investor_relations_news_company()` to access a specific company’s investor relations announcements.

2.3.4 Press releases from a specific company

Combine the methods `get_entity_search_results()` and `get_stories_specific_company()` to access press releases published by a specific company.

First, initialize the API and use `get_entity_search_results()` to search the API's database for any results matching the company you are looking for. For example the company "ARD":

```
>>> from pypresseportal import PresseportalApi
>>> api_object = PresseportalApi(YOUR_API_KEY)
>>> search_results = api_object.get_entity_search_results(search_term=["ARD"], entity=
↳ "company")
```

Next, inspect the search results. `get_entity_search_results()` returns a list of all companies matching your search string. Note that `get_entity_search_results()` will return `None` if the API did not find any matching entries, so make sure to check first:

```
>>> if search_results: # Check if search yielded any results
>>>     for company in search_results:
>>>         print(company.id, company.name)
6694 ARD Das Erste
22512 ARD ZDF
29876 ARD Presse
64887 ARDEX GmbH
73846 ARD Das Erste / ZDF
(...)
```

Finally, pick the id of the company you were looking for and pass it to `get_stories_specific_company()`, using the attribute `id`:

```
>>> company_stories = api_object.get_stories_specific_company(id=search_results[0].id)
>>> for story in company_stories:
>>>     print(story.title)
>>>     print(story.body)
Das Erste / "Wenn Frauen Austern essen" - der erste Gewinner des "Tatort"-Votings zum
↳ 50-jährigen Jubiläum der Krimireihe (FOTO)
München (ots) - 143.997 Zuschauerinnen und Zuschauer aus Deutschland und Österreich
↳ beteiligten sich an der ersten Abstimmungsrunde des Sommer-Events.
(...)
```

2.3.5 Investor relations announcements from a specific company

presseportal.de keeps investor relations announcements by public companies separated from regular press releases. Combine the methods `get_entity_search_results()` and `get_investor_relations_news_company()` to access investor relations announcements from a specific company.

First, initialize the API and use `get_entity_search_results()` to search the API's database for any results matching the company you are looking for. For example the company "Fraport":

```
>>> from pypresseportal import PresseportalApi
>>> api_object = PresseportalApi(YOUR_API_KEY)
>>> search_results = api_object.get_entity_search_results(search_term=["Fraport"],
↳ entity="company")
```

Next, inspect the search results. `get_entity_search_results()` returns a list of all companies matching your search string. Note that `get_entity_search_results()` will return `None` if the API did not find any matching entries, so make sure to check first:

```
>>> if search_results: # Check if search yielded any results
>>>     for company in search_results:
>>>         print(company.id, company.name)
31522 Fraport AG
```

Optionally, you can now use the method `get_company_information()` to query the API for more information about the company, such as **WKN**, **ISIN** or the company's RSS feed:

```
>>> company_info = api_object.get_company_information(id=search_results[0].id)
>>> print(company_info.wkn)
>>> print(company_info.isin)
>>> print(company_info.rss)
577330
DE0005773303
https://www.presseportal.de/rss/pm_31522.rss2
```

Finally, use the `get_investor_relations_news_company()` method with the `id` you acquired above to access the company's most recent investor relations press releases:

```
>>> investor_relations_stories = api_object.get_investor_relations_news_
↳company(search_results[0].id)
>>> for story in investor_relations_stories:
>>>     print(story.title)
>>>     print(story.body)
EANS-Hinweisbekanntmachung: Fraport AG Frankfurt Airport Services Worldwide /
Bekanntmachung gemäß § 37v, 37w, 37x ff. WpHG mit dem Ziel der europaweiten
Verbreitung
Hiermit gibt die Fraport AG Frankfurt Airport Services Worldwide bekannt,
(...)
```

2.4 Modules Documentation

2.4.1 The pypresseportal module

The `PresseportalApi` class

class `pypresseportal.PresseportalApi` (*api_key: str*)

A Python interface into the presseportal.de API.

The website presseportal.de is a service provided by 'news aktuell', owned by dpa (Deutsche Presse Agentur). An API key from presseportal.de is required. You can find more information and apply for an API key at <https://api.presseportal.de/en>. The documentation for PyPresseportal is at <https://pypresseportal.readthedocs.io/>.

First, create an instance of the `PresseportalApi` class, using your API key:

```
>>> from pypresseportal import PresseportalApi
>>> api_object = PresseportalApi(YOUR_API_KEY)
```

Next, request data from the API, using the various methods in this class.

For example:

Use the method `get_stories()` to access the most recently published stories (<https://api.presseportal.de/doc/article/all>). If no arguments are provided, PyPresseportal defaults to retrieving the 50 most recent stories available. For example:

```
>>> stories = api_object.get_stories()
```

Story data can be accessed through the individual Story object's attributes. For example:

```
>>> stories[0].title
"Kohls Wohnhaus hat keinen Denkmalwert"
>>> stories[0].id
"4622388"
```

get_company_information (*id: str*) → `pypresseportal.pypresseportal.Company`
Queries API for detailed information about a specific company.

Returns a `pypresseportal.Company` object. More information: <https://api.presseportal.de/doc/info/company/id>

Parameters *id* (*str*) – id of company (read `Entity.id` of a `get_entity_search_results()` search for this id).

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.

Returns An object containing details about the requested company.

Return type `Company`

get_entity_search_results (*search_term: Union[str, List[str]]*, *entity: str = 'company'*, *limit: int = 20*) → `Optional[List[pypresseportal.pypresseportal.Entity]]`
Search for company or public service office by location or name.

Returns a list of `pypresseportal.Entity` objects. More information: <https://api.presseportal.de/doc/search/company>

Parameters

- **search_term** (*Union[str, List[str]]*) – One or multiple search terms (min 3 chars.).
- **entity** (*str, optional*) – Search for 'office' or 'company'. Defaults to "company".
- **limit** (*int, optional*) – Limit number of objects in response (API maximum is 20). Defaults to 20.

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.
- **SearchTermError** – Api does not support the requested search term.
- **SearchEntityError** – Api does not support the requested entity.

Returns List of Entity objects. None if nothing found.

Return type `Union[List[Entity], None]`

get_investor_relations_news (*news_type: str = 'all'*, *start: int = 0*, *limit: int = 50*, *teaser: bool = False*) → `List[pypresseportal.pypresseportal.Story]`
Queries API for most recent investor relations press releases.

Returns a list of `pypresseportal.Story` objects. More information: <https://api.presseportal.de/doc/ir/list>

Parameters

- **news_type** (*str, optional*) – Investor relations news type (https://api.presseportal.de/doc/value/ir_type). Defaults to “all”.
- **start** (*int, optional*) – Start/offset of the result article list. Defaults to 0.
- **limit** (*int, optional*) – Limit number of articles in response (API maximum is 50). Defaults to 50.
- **teaser** (*bool, optional*) – Returns stories with `teaser` instead of `body` (fulltext) if set to True. Defaults to False.

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.
- **NewsTypeError** – API does not support the requested news type.

Returns List of Story objects

Return type List[*Story*]

get_investor_relations_news_company (*id: str, news_type: str = 'all', start: int = 0, limit: int = 50, teaser: bool = False*) → List[pypresseportal.pypresseportal.Story]

Queries API for investor relations press releases of a specific company.

Returns a list of `pypresseportal.Story` objects. More information: <https://api.presseportal.de/doc/ir/company/id/list>

Parameters

- **id** (*str*) – id of company (read `Entity.id` of a `get_entity_search_results()` search for this id).
- **news_type** (*str, optional*) – Investor relations news type (https://api.presseportal.de/doc/value/ir_type). Defaults to “all”.
- **start** (*int, optional*) – Start/offset of the result article list. Defaults to 0.
- **limit** (*int, optional*) – Limit number of articles in response (API maximum is 50). Defaults to 50.
- **teaser** (*bool, optional*) – Returns stories with `teaser` instead of `body` (fulltext) if set to True. Defaults to False.

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.
- **NewsTypeError** – API does not support the requested news type.

Returns List of Story objects

Return type List[*Story*]

get_public_service_news (*media: str = None, start: int = 0, limit: int = 50, teaser: bool = False*) → List[pypresseportal.pypresseportal.Story]

Queries API for public service news (police and fire departments, etc.).

Returns a list of `pypresseportal.Story` objects. More information: <https://api.presseportal.de/doc/article/publicservice>.

Parameters

- **media** (*str, optional*) – Only request stories containing this specific media type (image or document). Defaults to None.
- **start** (*int, optional*) – Start/offset of the result article list. Defaults to 0.
- **limit** (*int, optional*) – Limit number of articles in response (API maximum is 50). Defaults to 50.
- **teaser** (*bool, optional*) – Returns stories with `teaser` instead of `body` (fulltext) if set to True. Defaults to False.

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.
- **MediaError** – API does not support the requested media type.

Returns List of Story objects

Return type List[*Story*]

get_public_service_office_information (*id: str*) → `pypresseportal.pypresseportal.Office`
Queries API for detailed information about a specific public service office (police or fire department, etc.).

Returns a `pypresseportal.Office` object. More information: <https://api.presseportal.de/doc/info/office/id>

Parameters **id** (*str*) – id of office (read `Entity.id` of a `get_entity_search_results()` search for this id).

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.

Returns An object containing details about the requested office.

Return type *Office*

get_public_service_specific_office (*id: str, media: str = None, start: int = 0, limit: int = 50, teaser: bool = False*) →
List[`pypresseportal.pypresseportal.Story`]

Queries API for stories released by a specific public service office.

Returns a list of `pypresseportal.Story` objects. More information: <https://api.presseportal.de/doc/article/publicservice/office/id>

Parameters

- **id** (*str*) – id of office (read `Entity.id` of a `get_entity_search_results()` search for this id).
- **media** (*str, optional*) – Only request stories containing this specific media type (image or document). Defaults to None.
- **start** (*int, optional*) – Start/offset of the result article list. Defaults to 0.
- **limit** (*int, optional*) – Limit number of articles in response (API maximum is 50). Defaults to 50.

- **teaser** (*bool, optional*) – Returns stories with `teaser` instead of `body` (fulltext) if set to `True`. Defaults to `False`.

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.
- **MediaError** – API does not support the requested media type.

Returns List of Story objects

Return type List[*Story*]

get_public_service_specific_region (*region_code: str, media: str = None, start: int = 0, limit: int = 50, teaser: bool = False*) → List[pypresseportal.pypresseportal.Story]

Queries API for stories by public service offices in a specific region.

Returns a list of *pypresseportal.Story* objects. List of region codes and more information: <https://api.presseportal.de/doc/article/publicservice/region>.

Parameters

- **region_code** (*str*) – Only request stories located in this specific region.
- **media** (*str, optional*) – Only request stories containing this specific media type (image or document). Defaults to `None`.
- **start** (*int, optional*) – Start/offset of the result article list. Defaults to 0.
- **limit** (*int, optional*) – Limit number of articles in response (API maximum is 50). Defaults to 50.
- **teaser** (*bool, optional*) – Returns stories with `teaser` instead of `body` (fulltext) if set to `True`. Defaults to `False`.

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.
- **MediaError** – API does not support the requested media type.
- **RegionError** – API does not support the requested region code.

Returns List of Story objects

Return type List[*Story*]

get_stories (*media: str = None, start: int = 0, limit: int = 50, teaser: bool = False*) → List[pypresseportal.pypresseportal.Story]

Queries API for most recent press releases.

Returns a list of *pypresseportal.Story* objects. More information: <https://api.presseportal.de/doc/article/all>.

Parameters

- **media** (*str, optional*) – Only request stories containing this specific media type (image, document, audio or video). Defaults to `None`.
- **start** (*int, optional*) – Start/offset of the result article list. Defaults to 0.
- **limit** (*int, optional*) – Limit number of articles in response (API maximum is 50). Defaults to 50.

- **teaser** (*bool, optional*) – Returns stories with `teaser` instead of `body` (fulltext) if set to `True`. Defaults to `False`.

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.
- **MediaError** – API does not support the requested media type.

Returns List of Story objects

Return type List[*Story*]

get_stories_keywords (*keywords: List[str], media: str = None, start: int = 0, limit: int = 50, teaser: bool = False*) → List[pypresseportal.pypresseportal.Story]

Queries API for most recent press releases assigned to specific keywords.

Returns a list of *pypresseportal.Story* objects. More information: <https://api.presseportal.de/doc/article/keyword/ident>

Parameters

- **keywords** (*List[str]*) – A list of one ore more keywords from <https://api.presseportal.de/doc/value/keyword>
- **media** (*str, optional*) – Only request stories containing this specific media type (image, document, audio or video). Defaults to `None`.
- **start** (*int, optional*) – Start/offset of the result article list. Defaults to `0`.
- **limit** (*int, optional*) – Limit number of articles in response (API maximum is `50`). Defaults to `50`.
- **teaser** (*bool, optional*) – Returns stories with `teaser` instead of `body` (fulltext) if set to `True`. Defaults to `False`.

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.
- **MediaError** – API does not support the requested media type.
- **KeywordError** – API does not support the requested keyword(s).

Returns List of Story objects

Return type List[*Story*]

get_stories_specific_company (*id: str, media: str = None, start: int = 0, limit: int = 50, teaser: bool = False*) → List[pypresseportal.pypresseportal.Story]

Queries API for press releases of a specific company.

Returns a list of *pypresseportal.Story* objects. More information: <https://api.presseportal.de/en/doc/article/company/id>

Parameters

- **id** (*str*) – id of company (read `Entity.id` of a `get_entity_search_results()` search for this id).
- **media** (*str, optional*) – Only request stories containing this specific media type (image, document, audio or video). Defaults to `None`.

- **start** (*int, optional*) – Start/offset of the result article list. Defaults to 0.
- **limit** (*int, optional*) – Limit number of articles in response (API maximum is 50). Defaults to 50.
- **teaser** (*bool, optional*) – Returns stories with `teaser` instead of `body` (fulltext) if set to True. Defaults to False.

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.
- **MediaError** – API does not support the requested media type.

Returns List of Story objects

Return type List[*Story*]

get_stories_topic (*topic: str, media: str = None, start: int = 0, limit: int = 50, teaser: bool = False*) → List[pypresseportal.pypresseportal.Story]

Queries API for most recent press releases assigned to a specific topic.

Returns a list of *pypresseportal.Story* objects. More information: <https://api.presseportal.de/doc/article/topic/ident>

Parameters

- **topic** (*str*) – One specific topic from <https://api.presseportal.de/doc/value/topic>
- **media** (*str, optional*) – Only request stories containing this specific media type (image, document, audio or video). Defaults to None.
- **start** (*int, optional*) – Start/offset of the result article list. Defaults to 0.
- **limit** (*int, optional*) – Limit number of articles in response (API maximum is 50). Defaults to 50.
- **teaser** (*bool, optional*) – Returns stories with `teaser` instead of `body` (fulltext) if set to True. Defaults to False.

Raises

- **ApiConnectionFail** – Could not connect to API.
- **ApiError** – API returned an error.
- **MediaError** – API does not support the requested media type.
- **TopicError** – API does not support the requested topic.

Returns List of Story objects

Return type List[*Story*]

The Story class

class `pypresseportal.Story` (*data: dict*)

Represents a story retrieved through the API.

See original API documentation for details (<https://api.presseportal.de/en/doc/format/story>).

A `Story` object can contain different attributes, depending on the API query method:

- `data` - The raw json data returned by the API.
- `id` - The story's unique id as generated by the API.
- `url` - URL of the story on the `presseportal.de` website.
- `title` - Headline of the story.
- `body` - Full text body.
- `optional: teaser` - Teaser text. Only included if `teaser=True` is passed to the API query method. If a teaser is requested, it will replace the `body` attribute.
- `published` - Publication date (as datetime object).
- `language` - Language of the story (usually 'de' or 'en').
- `ressort` - Editorial department.
- `company_id/office_id` - Unique id of the company or office publishing the story.
- `company_url/office_url` - URL of the company or office publishing the story.
- `company_name/office_name` - Name of the company or office publishing the story.
- `keywords` - List of keywords assigned to the story.
- `optional: media` - Information on media attachments, if present.
- `highlight` - Promoted story flag (on/off).
- `short` - Shortened URL.

The Entity class

class `pypresseportal.Entity` (*data: dict*)

Represents a company or a public service office search result.

An `Entity` object can have the following attributes: `id`, `url`, `name`, and `type` (company or office). See original API documentation for details (<https://api.presseportal.de/doc/format/result>).

The Company class

class `pypresseportal.Company` (*data: dict*)

Represents information about a company.

A `Company` object can have the following attributes: `id`, `url`, `name`, `isin` (optional), `wkn` (optional), `shortname`, `rss`, `logo`, `web`, and `homepage`. See original API documentation for details (<https://api.presseportal.de/en/doc/format/company>).

The Office class

class `pypresseportal.Office` (*data: dict*)
Represents information about a public service office.

An Office object can have the following attributes: `id`, `url`, `name`, `shortname`, `rss`, `logo`, `web`, and `homepage`. See original API documentation for details (<https://api.presseportal.de/doc/format/office>).

2.4.2 The `pypresseportal_errors` module

Error handling for PyPresseportal.

exception `pypresseportal.pypresseportal_errors.ApiConnectionFail` (*error_msg: Union[requests.exceptions.ConnectionError, requests.exceptions.TooManyRedirects, requests.exceptions.Timeout]*)

Raised if `requests` raises an error.

Parameters `error_msg` (*Union[requests.exceptions.ConnectionError, requests.exceptions.TooManyRedirects, requests.exceptions.Timeout,]*) – Error raised by `requests` package.

exception `pypresseportal.pypresseportal_errors.ApiDataError` (*msg: str = None*)
Raised if the API returned invalid data.

Parameters `msg` (*str, optional*) – Custom error message. Defaults to `None`.

exception `pypresseportal.pypresseportal_errors.ApiError` (*error_code: str, error_msg: str*)

Raised if API returns an error.

Parameters

- **error_code** (*str*) – Error code returned by API.
- **error_msg** (*str*) – Error message returned by API.

exception `pypresseportal.pypresseportal_errors.ApiKeyError` (*api_key: str*)
Raised if no API key is provided.

Parameters `api` – key (*str*): Invalid API key.

exception `pypresseportal.pypresseportal_errors.KeywordError` (*keyword: str, allowed: tuple*)

Raised if keyword is not supported. List of keywords: <https://api.presseportal.de/en/doc/value/keyword>.

Parameters

- **keyword** (*str*) – Unsupported keyword.
- **allowed** (*list*) – List of keyword allowed by API.

exception `pypresseportal.pypresseportal_errors.MediaError` (*media: Optional[str], allowed: tuple*)

Raised if media type is not supported. List of media types: <https://api.presseportal.de/en/doc/value/media>.

Parameters

- **media** (*str*) – Unsupported media type.
- **allowed** (*list*) – List of media types allowed by API.

exception `pypresseportal.pypresseportal_errors.NewsTypeError` (*news_type: str, allowed: tuple*)

Raised if investor relations news type is not supported. List of news types: https://api.presseportal.de/en/doc/value/ir_type.

Parameters

- **news_type** (*str*) – Unsupported investor relations news type.
- **allowed** (*list*) – List of investor relations news type allowed by API.

exception `pypresseportal.pypresseportal_errors.RegionError` (*region: str, allowed: tuple*)

Raised if region is not supported. List of regions: <https://api.presseportal.de/en/doc/value/region>.

Parameters

- **region** (*str*) – Unsupported region.
- **allowed** (*list*) – List of regions allowed by API.

exception `pypresseportal.pypresseportal_errors.SearchEntityError` (*search_term: str*)

Raised if entity for search is not supported.

Parameters **entity** (*str*) – Unsupported entity.

exception `pypresseportal.pypresseportal_errors.SearchTermError` (*search_term: str*)

Raised if search term is not supported.

Parameters **search_term** (*str*) – Unsupported search term.

exception `pypresseportal.pypresseportal_errors.TopicError` (*topic: str, allowed: tuple*)

Raised if topic is not supported. List of topics: <https://api.presseportal.de/en/doc/value/topic>.

Parameters

- **topic** (*str*) – Unsupported topic.
- **allowed** (*list*) – List of topics allowed by API.

INDICES AND TABLES

- [genindex](#)
- [Modules Documentation](#)
- [search](#)

PYTHON MODULE INDEX

p

`pypresseportal.pypresseportal_errors,`

17

A

[ApiConnectionFail](#), 17
[ApiDataError](#), 17
[ApiError](#), 17
[ApiKeyError](#), 17

C

[Company](#) (*class in pypresseportal*), 16

E

[Entity](#) (*class in pypresseportal*), 16

G

[get_company_information\(\)](#) (*pypresseportal.PresseportalApi method*), 10
[get_entity_search_results\(\)](#) (*pypresseportal.PresseportalApi method*), 10
[get_investor_relations_news\(\)](#) (*pypresseportal.PresseportalApi method*), 10
[get_investor_relations_news_company\(\)](#) (*pypresseportal.PresseportalApi method*), 11
[get_public_service_news\(\)](#) (*pypresseportal.PresseportalApi method*), 11
[get_public_service_office_information\(\)](#) (*pypresseportal.PresseportalApi method*), 12
[get_public_service_specific_office\(\)](#) (*pypresseportal.PresseportalApi method*), 12
[get_public_service_specific_region\(\)](#) (*pypresseportal.PresseportalApi method*), 13
[get_stories\(\)](#) (*pypresseportal.PresseportalApi method*), 13
[get_stories_keywords\(\)](#) (*pypresseportal.PresseportalApi method*), 14
[get_stories_specific_company\(\)](#) (*pypresseportal.PresseportalApi method*), 14
[get_stories_topic\(\)](#) (*pypresseportal.PresseportalApi method*), 15

K

[KeywordError](#), 17

M

[MediaError](#), 17
 module
 [pypresseportal.pypresseportal_errors](#), 17

N

[NewsTypeError](#), 17

O

[Office](#) (*class in pypresseportal*), 17

P

[PresseportalApi](#) (*class in pypresseportal*), 9
[pypresseportal.pypresseportal_errors](#) module, 17

R

[RegionError](#), 18

S

[SearchEntityError](#), 18
[SearchTermError](#), 18
[Story](#) (*class in pypresseportal*), 16

T

[TopicError](#), 18