babelfish Documentation

Release 0.1.1

Antoine Bertin

September 27, 2013

CONTENTS

Release v0.1.1

BabelFish is a Python library to work with countries and languages.

CHAPTER

ONE

COUNTRY

Simple country representation from two-letters code (ISO-3166):

```
>>> country = babelfish.Country('GB')
>>> country
<Country UNITED KINGDOM>
```

CHAPTER

TWO

LANGUAGE

Simple language representation from three-letters code (ISO-639-3):

```
>>> language = babelfish.Language('eng')
>>> language
<Language English>
```

Country specific language:

```
>>> language = babelfish.Language('por', 'BR')
>>> language
<Language Portuguese, country=BRAZIL>
```

Built-in converters (alpha2, alpha3b, name and opensubtitles):

```
>>> language.alpha2
'pt'
>>> language.opensubtitles
'pob'
>>> babelfish.Language.fromalpha3b('fre')
<Language French>
```

CHAPTER

THREE

CUSTOM CONVERTERS

Build your own converter:

```
class MyCodeConverter(babelfish.ReverseConverter):
    def __init__(self):
        self.to_mycode = {'fra': 'mycode1', 'eng': 'mycode2'}
        self.from_mycode = {'mycode1': 'fra', 'mycode2': 'eng'}
    def convert(self, alpha3, country=None):
        if alpha3 not in self.to_mycode:
            raise babelfish.NoConversionError
        return self.to_mycode[alpha3]
    def reverse(self, mycode):
        if mycode not in self.from_mycode:
            raise babelfish.NoConversionError
        return (self.from_mycode[mycode], None)
```

Use it:

```
>>> babelfish.register_converter('mycode', MyCodeConverter)
>>> babelfish.Language.frommycode('mycode2')
<Language English>
>>> babelfish.Language('fra').mycode
'mycode1'
```

CHAPTER

FOUR

API DOCUMENTATION

If you are looking for information on a specific function, class or method, this part of the documentation is for you.

4.1 Country

`babelfish.country.`**`COUNTRIES`**
Dictionary of country ISO-3166 codes to English names

class `babelfish.country.`**`Country`**(*country*)
A country on Earth

A country is represented by a two-letters code from the ISO-3166 standard

Parameters **country** (*string*) – two-letters ISO-3166 country code

`alpha2` = **None**
ISO-3166 two-letters country code

`name`
English name of the country

4.2 Language

`babelfish.language.`**`LANGUAGES`**
Set of ISO-639-3 three-letters code

class `babelfish.language.`**`Language`**(*language*, *country=None*)
A human language

A human language is composed of a language part following the ISO-639 standard and can be country-specific when a `Country` is specified.

The `Language` is extensible with custom converters (see *Custom Converters*)

Parameters

- **language** (*string*) – the language as a three-letters ISO-639-3 code
- **country** (string or `Country` or None) – the country (if any) as a two-letters ISO-3166 code or `Country` instance

`babelfish.language.`**`CONVERTERS`**
Registered converters

`babelfish.language.`**`register_converter`**(*name*, *converter*)

Register a `Converter` with the given name

This will add the *name* property to the `Language` class and an alternative constructor *fromname* if the *converter* is a `ReverseConverter`

Parameters

- **name** (*string*) – name of the converter to register
- **converter** (`Converter`) – converter to register

`babelfish.language.`**`unregister_converter`**(*name*)

Unregister a `Converter` by name

Parameters **name** (*string*) – name of the converter to unregister

`babelfish.language.`**`load_converters`**()

Load converters from the entry point

Call `register_converter()` for each entry of the 'babelfish.converters' entry point

`babelfish.language.`**`clear_converters`**()

Clear all converters

Call `unregister_converter()` on each registered converter in `CONVERTERS`

4.3 Converter Bases

class `babelfish.converters.`**`Converter`**

A `Converter` supports converting an alpha3 language code with an alpha2 country code into a custom code

`convert`(*alpha3*, *country=None*)

Convert an alpha3 language code with an alpha2 country code into a custom code

Parameters

- **alpha3** (*string*) – ISO-639-3 language code
- **country** (*string or None*) – ISO-3166 country code, if any

Returns the corresponding custom code

Return type string

class `babelfish.converters.`**`ReverseConverter`**

A `Converter` able to reverse a custom code into a alpha3 ISO-639-3 language code and alpha2 ISO-3166-1 country code

`reverse`(*code*)

Reverse a custom code into alpha3 and country code

Parameters **code** (*string*) – custom code to reverse

Returns the corresponding alpha3 ISO-639-3 language code and alpha2 ISO-3166-1 country code

Return type tuple

4.4 Exceptions

class `babelfish.exceptions.`**`BabelfishError`**

Base class for all exceptions in babelfish

class `babelfish.exceptions.`**`NoConversionError`**

Raised when no conversion could be done with the converter on the given code

CHAPTER

FIVE

BABELFISH CHANGELOG

5.1 0.1

Release date: TBA

- Initial version

PYTHON MODULE INDEX

b

babelfish.converters, **??**
babelfish.country, **??**
babelfish.exceptions, **??**
babelfish.language, **??**