HER Documentation

Release 1.4.0

hearot

Mar 12, 2019

Contents

sphinx-quickstart on Mon Apr 16 21:03:36 2018. You can adapt this file completely to your liking, but it should at least contain the root *toctree* directive.

CHAPTER 1

her

It contains the main functions used to convert an HER string to a dictionary and viceversa.

CHAPTER 2

her

2.1 her package

2.1.1 Submodules

2.1.2 her.decoder module

her.decoder

It contains the decoder functions.

It could be used to decode HER and convert it into a dictionary.

her.decoder.**decode**(*representation*)
: It parses the string and convert it into a dictionary.

  Parameters **representation** (*list*) – A list of the lines of the string.

  Return type Dict[str, Any]

2.1.3 her.encoder module

her.encoder

It contains the encoder functions.

It could be used to encode a dictionary and convert it into an HER text.

her.encoder.**encode**(*dictionary*)
: It parses and convert the dictionary into an HER text.

  Parameters **dictionary** (*dict*) – The dictionary you want to convert.

  Return type str

2.1.4 her.her module

her.core.her

It contains all objects that represents an HER file.

class her.her.**HER**(*value=None*)
Bases: object

The HER class is used to avoid multiple calls of the encode & decode functions. It uses properties, so it updates all its attributes every single time you assign a new value to an attribute.

Example:

```
x = her.HER()
x.value = {"foo": {"lol": 1}}
print(repr(x.representation)) # Output: '- foo -\n    * lol = 1'
```

You can also pass a parameter (dict or str):

```
x = her.HER('- foo -\n    * lol = 1')
print(x.value) # Output: {"foo": {"lol": 1}}

y = her.HER({"foo": {"lol": 1}})
print(repr(x.representation)) # Output: '- foo -\n    * lol = 1'
```

representation
The string which represents the set value using HER standards.

Returns The representation.

Return type str

value
The value which represents the content of the HER string as a dictionary.

Returns The value.

Return type dict

2.1.5 Module contents

her

It contains the main functions used to convert an HER string to a dictionary and viceversa.

CHAPTER 3

HER, a new Text Format

Search informations about the Syntax and Types using the Wiki section.

3.1 Content table

3.2 What's HER?

HER is text format, like XML/Json. The difference is that HER is easier than others. Just see:

```
- Category -
    >> Array[]
    * Array[] = "Umh, that's pretty good!"
```

3.3 Why shall I use HER?

As I said before, HER is **simple** and **easy to use**. You can pass informations, or better, store informations* and document them.

Feel the difference:

XML:

```
<christmas>
    <greetings>Merry christmas!</greetings>
    <greetings>Spam, Python, Eggs</greetings>
</christmas>
```

HER:

```
- Christmas -
    >> Greetings[]
    * Greetings[] = "Merry christmas!"
    * Greetings[] = "Spam, Python, Eggs"
```

CHAPTER 4

Python Module

4.1 Installation

You can easily install that module using pip:

```
pip install her
```

Or, if you want to upgrade the module:

```
pip install --upgrade her
```

4.2 Import Module

You must use `import her` to import all the HER module.

```
import her
...
```

4.3 Encode a Dictionary

Just use the `encode` function.

```
from her import encode
her = encode({'Category':{'hello world':True}})
print(her)
```

Output:

```
- Category -
    * hello world = True
```

4.4 Decode a String

Just use the `decode` function.

```
from her import decode
dictionary = decode("- Category -\n    * hello world = True")
print(dictionary)
```

Output:

```
{'Category':{'hello world':True}}
```

4.5 HER class

You can use the `HER` class to call less `encode` & `decode` functions and optimize your codebase. It updates all its attributes automatically.

```
x = her.HER()
x.value = {"foo": {"lol": 1}}
print(repr(x.representation)) # Output: '- foo -\n    * lol = 1'
```

You can also pass a parameter (*dict* or *str*):

```
x = her.HER('- foo -\n    * lol = 1')
print(x.value) # Output: {"foo": {"lol": 1}}

y = her.HER({"foo": {"lol": 1}})
print(repr(x.representation)) # Output: '- foo -\n    * lol = 1'
```

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

h

Index