HER Documentation

Release 1.4.0

hearot

Nov 01, 2018

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

1.1 her package

1.1.1 Submodules

1.1.2 her.decoder module

1.1.3 her.encoder module

1.1.4 her.her module

1.1.5 Module contents

CHAPTER 2

HER, a new Text Format

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

2.1 Content table

2.2 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!"
```

2.3 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:

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

HER:

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

2.3.1 Why HER?

Because, however, if it isn't right, it's right. (just joking)

CHAPTER 3

Python Module

3.1 Installation

You can easily install that module using PiP:

```
pip install her
```

Or, if you want to upgrade the module:

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

3.2 Import Module

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

```
import her
...
```

3.3 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
```

3.4 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}}
```

3.5 HER class

You can use the `HER` class to call less `encode` & `decode` functions an 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 4

Indices and tables

- genindex
- modindex
- search