WireCurly Documentation

Release 0.1

Indicium SRL

December 02, 2013

Contents

`WireCurly` is a project sponsored by Indicium SRL and it's main objective is assist developers write configuration for *FreeSWITCH*™.

`WireCurly` is currently in alpha stage although it is being intensively used in production systems. The main reason why it is in alpha stage is because it is incomplete and only the features needed by specific vendor applications are implemented. We will soon complete the lib open it's official release annoucement.

CHAPTER 1

Project guidelines

`WireCurly` is developed with the developer in mind. *FreeSWITCH*™ configuration is XML based and parsing/writing XML using templates can be very error prone due to repetitive tasks. This is where WireCurly excels providing a very pythonic interface for writing all parts of *FreeSWITCH*™.

CHAPTER 2

Examples

Here's a little example of how to write a simple directory entry:

```
from wirecurly.directory import User

user = User("1000", "1234")
user.addParameter('vm-password', "1234")
user.addVariable('user_context', 'default')
```

To create simple dialplan entries, the same thing applies. Here's an example:

```
from wirecurly.dialplan import Extension
from wirecurly.dialplan.condition import *
from wirecurly.dialplan.expression import *
from wirecurly.dialplan.applications.bridge import Bridge

ext = Extension('simple_extension')

conda = Condition(expr=ExpressionField("destination_number", "^1000$"))
conda.addAction(Bridge("user/1000"))

condb = Condition(expr=ExpressionField("destination_number", "^2000$"))
condb.addAction(Bridge("user/20000"))

ext.addCondition(or_(conda, condb))
```

CHAPTER 3

Serialization

Currently, only file serialization is implemented based on `XMLFileFactory`.

CHAPTER 4

Complete list of modules

modindex

See Also:

`glossary`