resourceSet module¶

`resourceSet.RS_from_xml`(*tree*)¶

function for xml parsing which return a ResourceSet

Parameters: **tree** (*ElementTree*) – tree is a node of a resourceSet
Returns: Resource creatinf
Return type: ResourceSet

`resourceSet.R_toxml`(*r*)¶

function for parsing node into a xml

Parameters: **r** (*Resource*) – Resource
Returns: a node from Element tree
Return type: Element tree

class `resourceSet.Resource`(*typ=None*, *prop=None*, *name=None*, *host=None*)¶

Bases: `object`

Class Resource

Attributes:
type type of the resource of type string properties properties of the resources of type dict

`__dict__` = *dict_proxy({'gw': <function gateway at 0x7fde24e10b18>, 'make_taktuk_command': <function*

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

make_taktuk_command at 0x7fde24e10c80>, '__module__': 'resourceSet', '__str__': <function __str__ at 0x7fde24e10758>, 'corresponds': <function corresponds at 0x7fde24e10938>, 'name_equal': <function name_equal at 0x7fde24e107d0>, 'job': <function job at 0x7fde24e10c08>, 'gw_ssh_user': <function gw_ssh_user at 0x7fde24e108c0>, 'gateway_equal': <function gateway_equal at 0x7fde24e10b90>, '__dict__': <attribute '__dict__' of 'Resource' objects>, 'copy': <function copy at 0x7fde24e109b0>, '__eq__': <function __eq__ at 0x7fde24e10a28>, 'gateway': <function gateway at 0x7fde24e10b18>, '__init__': <function __init__ at 0x7fde24e93c80>, 'name': <function name at 0x7fde24eae050>, '__weakref__': <attribute '__weakref__' of 'Resource' objects>, 'host': <function host at 0x7fde24e10cf8>, 'ssh_user': <function ssh_user at 0x7fde24e10848>, 'eql': <function eql at 0x7fde24e10aa0>, '__doc__': '\n Class Resource \n\n Attributes:\n type type of the resource of type string\n properties properties of the resources of type dict\n\n '})¶

`__eq__`(*res*)¶

Equality, Two Resource objects are equal if they have the same type and the same properties as well.

Returns: result of the test
Return type: boolean

`__init__`(*typ=None*, *prop=None*, *name=None*, *host=None*)¶

Creates a new Resource Object.

Parameters:

- **typ** (*string*) – the type of the object could be “node” or “Resource_set”
- **prop** (*dict*) – object property
- **name** (*string*) – String name
- **host** (*execo.host*) – Host to convert

Returns:

Resource Object

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Return type:
Resource

Example:

```
>>> r = Resource("node",name="toto")
<resourceSet.Resource object at .... >
```

`__module__` = *'resourceSet'*¶

`__str__`()¶

Return the name of the resource.

Returns: the name of the resource
Return type: str

call with str(obj)

Example:

```
>>> str(r)
toto
```

`__weakref__`¶

list of weak references to the object (if defined)

`copy`()¶

Creates a copy of the resource object.

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Returns: copy of the resource
Return type: Resource object

`corresponds`(*props*)¶

check if a resource corresponds to the props

Also test whenever a value of the properties are callable: test the callable value with key as an argument

Parameters: **prop** (*dict*) – object property
Returns: True if the resource have the same properties than the parameters
Return type: Bool

`eql`(*res*)¶

Equality, Two Resource objects are equal if they have the same type and the same properties as well.

Returns: true if self and other are the same object.
Return type: boolean

`gateway`()¶

Returns the name of the gateway

Returns: Returns the name of the gateway
Return type: string

`gateway_equal`(*host*)¶

set the getaway

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Parameters: **host** (*string*) – host to be set

`gw`()¶

Returns the name of the gateway

Returns: Returns the name of the gateway
Return type: string

`gw_ssh_user`()¶

Return propertie gw_ssh_user.

Returns: properties gw_ssh_user.

`host`()¶

try to convert the ressource as an execo host use the properties : gateway, user, keyfile and port default gateway is localhost.

Returns: an execo Host
Return type: execo.Host

`job`()¶

get the id of the resource

Returns: Returns id
Return type: int

`make_taktuk_command`(*cmd*)¶

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Use to make the list of machines for

Returns: the taktuk command
Return type: string

`name`()¶

Return the name of the resource.

Returns: the name of the resource
Return type: str

`name_equal`(*name*)¶

Sets the name of the resource.

Parameters: name (*string*) – String name

`ssh_user`()¶

Return propertie ssh_user.

Returns: propertie ssh_user.

class `resourceSet.ResourceSet`(*name=None*)¶

Bases: `resourceSet.Resource`

Class ResourceSet heret from Resource

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Attributes:

type (inheret from Resource) type of the resource of type string properties (inheret from Resource) properties of the resources of type dict Resource list of the resources resrouces_files

`__eq__`(*set*)¶

test if current set are equal with parameter

Parameters: **set** (*ResourceSet or Resource*) –
Returns: result of the test
Return type: boolean

Note

can be called the python way with ==

`__getitem__`(*index*)¶

Returns a subset of the ResourceSet.

Parameters: **index** – [Range] Returns a subset specified by the range, [String] index Returns a subset which is belongs to the same cluster, [Integer] Returns just one resource .
Returns: a ResourceSet object
Return type: ResourceSet
Example:

```
>>> all[range(1,6)] #extract resources from 1 to 5
>>> all["lyon"]  #extract the resources form lyon cluster
>>> all[0] #return just one resource.
```

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Note

It can be used with a range(or int list) as a parameter or a string .

Warning

Raise StopIteration if call with a wrong parameter or looking for an inexistant resource.

`__init__`(*name=None*)¶

Creates a new ResourceSet Object.

Parameters: **name** (*string*) – String name
Returns: Resourceset Object
Return type: ResourceSet
Example:

```
>>> r = ResourceSet("node",name="toto")
<resourceSet.ResourceSet object at .... >
```

`__len__`()¶

Returns the number of node in the ResourceSet

can be called the python way : len(resourceSet)

Returns: the number of resources
Return type: Integer

`__module__` = *'resourceSet'*¶

`__ne__`(*set*)¶

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

test if current set are non equal with parameter

Parameters: **set** (*ResourceSet or Resource*) –
Returns: result of the test
Return type: boolean

Note

can be call the python way with !=

`append`(*resource*)¶

Add a Resource object to the ResourceSet

Parameters: **resource** – the resource or ResourceSet to add to the atributes resources of the current resourceSet

:type resource : Objet (can be Resource or ResourceSet) named append to stick with the usel python function

`copy`()¶

Creates a copy of the ResourceSet Object :return: Resourceset Object :rtype: ResourceSet

`delete`(*resource*)¶

delete all resources equal to the parameters resource return None if it has failed to found one

Parameters: **resource** (*ResourceSet or Resource*) – the resource which have to be deleted
Returns: the last resource deleted or None if none found

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Return type: Resource

`delete_first`(*resource*)¶

delete the first resource equal to the parameters resource return None if it has failed to found the resource

Parameters: **resource** (*ResourceSet or Resource*) – the resource which have to be deleted
Returns: the resource or None if none found
Return type: Resource

`delete_first_if`(*block=None*)¶

delete the first resource which return True with the given function in the block parameter return None if it has failed to found the resource

Parameters: **block** (*callable object*) – function or lambda function which take a resource in parameter and return Boolean
Returns: the resource or None if none found
Return type: Resource

`delete_if`(*block=None*)¶

delete all resource which return True with the given function in the block parameter return None if it has failed to found the resource

Parameters: **block** (*callable object*) – function or lambda function which take a resource in parameter and return Boolean
Returns: the last resource or None if none found

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Return type: Resource

each(*type=None*, *block=None*)¶

Browse the resourceSet .
This goes until the size of the ResourceSet.

Parameters:

- **type** (*string*) – the type of the object could be either “node” or “Resource_set”
- **block** (*callable object*) – function or lambda function which will be call on every resource

Note

this kind of resource browsing stick more with ruby functionnality than python one because Bloc are limited in python prefer a “for resource in ResourceSetIterator(self, “node”) ” to browse every node

each_slice(*type=None*, *slice_step=1*, *block=None*)¶

Creates groups of increasing size based on the slice_step paramater.
This goes until the size of the ResourceSet.

Parameters:

- **type** (*string*) – the type of the object could be either “node” or “Resource_set”
- **slice_step** (*int or function*) – int or function or lambda function which explicit the way how resources are browsed
- **block** (*callable object*) – function or lambda function which will be call on every resource

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Note

this kind of resource browsing stick more with ruby functionnality than python one because Bloc are limited in python

`each_slice_array`(*slices=1*, *block=None*)¶

`each_slice_double`(*type=None*, *block=None*)¶

Browse the resourceSet two per two .
This goes until the size of the ResourceSet.

Parameters:

- **type** (*string*) – the type of the object could be either “node” or “Resource_set”
- **block** (*callable object*) – function or lambda function which will be call on every resource

Note

this kind of resource browsing stick more with ruby functionnality than python one because Bloc are limited in python

`each_slice_power2`(*type=None*, *block=None*)¶

Browse the resourceSet exponentialy .
This goes until the size of the ResourceSet.

Parameters:

- **type** (*string*) – the type of the object could be either “node” or “Resource_set”
- **block** (*callable object*) – function or lambda function which will be call on every

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

resource

Note

this kind of resource browsing stick more with ruby functionnality than python one because Bloc are limited in python

`eql`(*set*)¶

Equality between to resoruce sets.

Parameters: set (*ResourceSet or Resource*) –
Returns: result of the test
Return type: boolean

`first`(*type=None*)¶

Return the first element which is an object of the Resource Class

Parameters: type (*string*) – the type of the object could be “node” or “Resource_set”
Returns:
Return type: Resource

`flatten`(*type=None*)¶

Puts all the resource hierarchy into one ResourceSet.

Parameters: type (*string*) – the type of the object could be either “node” or “Resource_set”
Returns: the ResourceSet with all resource faltened

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Return type: ResourceSet

`flatten_not`(*type=None*)¶

flatten the current resourceSet

Parameters: type (*string*) – the type of the object could be either “node” or “Resource_set”

See also

flatten

`hosts`()¶

try to convert the resourceSet as a list of execo host use the properties : gateway, user, keyfile and port default gateway is localhost.

Returns: a list of execo Host
Return type: list of execo.Host

`make_taktuk_command`(*cmd*)¶

Creates the taktuk command

Creates the taktuk command to execute on the ResourceSet It takes into account if the resources are grouped under different gatways in order to perform this execution more efficiently.

..todo:: verify this function

`select`(*type=None*, *props=None*, *block=None*)¶

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

select every resource of the given type which correspond to the props or the block (see warnings) and wrap it in a ResourceSet

Parameters:

- **type** (*string*) – the type of the object could be “node” or “Resource_set”
- **props** (*dict*) – props which the ressource should correspond to
- **block** (*function or lambda function which return boolean*) – function or lambda function which return boolean

Returns:

return a ResourceSet with all the resources which are selected

Return type: Resource

Example:

```
>>> r2 = r.select('node',block = (lambda x : x.name()=='tutu'))
>>> r1 = r.select('node',({'name': 'tutu'}))
r1 = r2
```

See also

correspond

Warning

if both parameters props and block are given only props will be treated

`select_resource`(*props*)¶

Return the first element which correspond to the properties given in parameters

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Parameters: **props** (*string*) – props which the ressource should correspond to
Returns: return the selected Resource Object
Return type: Resource

`to_resource`()¶

Returns a resouce or an array of resources.

Returns: a resource or array of resources
Return type: Resource

`uniq`()¶

Returns a ResourceSet with unique elements.

Returns: ResourceSet with unique elements
Return type: ResourceSet

`uniq_aux`()¶

Returns same ResourceSet with unique elements.

Returns: same ResourceSet with unique elements
Return type: ResourceSet

class `resourceSet.ResourceSetIterator`(*resource_set*, *type=None*)¶

Bases: `object`

Class ResourceSet heret from Resource

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

it will look in every resourceSet contained in the initial resourceSet usefull to make a resourceSet iterable see example

Attributes:

current : index for the ‘(<current element iterator : a resourceSet element to browset element from a sub resourceSet resource_set: initial resoureSet type : type of the init resourceSet

Example:

```
>>> from resourceSet import *
>>> resource_set = parser_xml("resourceSet.xml")
>>> for i in ResourceSetIterator(resource_set :
>>>     print i
node1
node2
node3
node4
```

`__init__`(*resource_set*, *type=None*)¶

Creates a new Resource Object.

Parameters:

- **resource_set** (*dict*) – resourceset on which the iterator will be created
- **typ** (*string*) – the type of the object could be “node” or “Resource_set”

Returns:

Resource Object

Return type:

Resource

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Note

if no type are given it will give every kind of object contained

`__iter__`()¶

method to make ResourceSetIterator iterable

`__module__` = *'resourceSet'*¶

`next`()¶

place the index current on the next element and return it

Returns: the next resource
Return type: resource of the same type as the attributes type
Example:

it = ResouceSetIterator(resource_set,"node") try : it.next() except StopIteration : ...

Warning

Raise a Stopiteration at the end

`resource`()¶

Give the current resource

Returns: the current resource
Return type: resource of the same type as the attributes type

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

`resourceSet.Rs_toxml`(*rs*)¶

function for parsing node into a xml

Parameters: **r** (*ResourceSet*) – ResourceSet
Returns: a node from Element tree
Return type: Element tree

`resourceSet.parser_xml`(*path*)¶

function to creat a ResrouceSet from an xml file

Parameters: **path** (*string*) – the path of the xml file
Returns: the ResourceSet created
Return type: ResourceSet
Example:

```
>>> parser_xml('xml/rs1.xml')
 <resourceSet.Resource object at .... >
```

`resourceSet.prettify`(*elem*)¶

Return a pretty-printed XML string for the Element. function taken from the site : https://pymotw.com/2/xml/etree/ElementTree/create.html

`resourceSet.res_from_xml`(*tree*)¶

function for xml parsing which return a Resource

Parameters: **tree** (*ElementTree*) – tree created with element tree
Returns: Resource creatinf

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Return type: Resource

`resourceSet.xml_writer`(*r*)¶

function to create xml data from a node

Parameters: **r** (*ResrousceSEt*) – ResrouceSet
Returns: the xml created

Related Topics

- Documentation overview

This Page

- Show Source

©2017, Timothée Lemaire - Nicolas Homberg. | Powered by Sphinx 1.3.6 & Alabaster 0.7.7 | Page source

PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com