

# f5.bigip.tm.util

## Module contents

BIG-IP® Utility (util) module

### REST URI

`http://localhost/mgmt/tm/util/`

### GUI Path

`System`

### REST Kind

N/A – HTTP GET returns an error

## Submodule List

<code>bash</code>	BIG-IP® utility module
<code>clientssl_ciphers</code>	BIG-IP® utility module
<code>dig</code>	BIG-IP® utility module
<code>get_dossier</code>	BIG-IP® utility module
<code>qkview</code>	BIG-IP® utility module
<code>serverssl_ciphers</code>	BIG-IP® utility module
<code>unix_ls</code>	BIG-IP® utility module
<code>unix_mv</code>	BIG-IP® utility module
<code>unix_rm</code>	BIG-IP® utility module

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

**class** `f5.bigip.tm.util.Util(bigip)` [\[source\]](#)

Bases: `f5.bigip.resource.PathElement`

**raw**

Display the attributes that the current object has and their values.

Returns: A dictionary of attributes and their values

## Submodules

### bash

BIG-IP® utility module

#### REST URI

`http://localhost/mgmt/tm/util/bash`

#### GUI Path

N/A

#### REST Kind

`tm:util:bash:*`

**class** `f5.bigip.tm.util.bash.Bash(util)` [\[source\]](#)

Bases: `f5.bigip.resource.UnnamedResource`,  
`f5.bigip.mixins.CommandExecutionMixin`

BIG-IP® utility command

#### 📌 Note

This is an unnamed resource so it has no ~Partition~Name pattern at the end of its URI.

**exec\_cmd**(*command*, **\*\*kwargs**) [\[source\]](#)

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

Wrapper method that can be changed in the inheriting classes.

#### **create(\*\*kwargs)**

Create is not supported for unnamed resources

##### **Raises:**

UnsupportedMethod: If the method is used

#### **delete(\*\*kwargs)**

Delete is not supported for unnamed resources

##### **Raises:**

UnsupportedMethod: If the method is used

#### **load(\*\*kwargs)**

Load is not supported for command execution

**Raises:**      UnsupportedOperation

#### **modify(\*\*patch)**

Modify the configuration of the resource on device based on patch

#### **raw**

Display the attributes that the current object has and their values.

**Returns:**      A dictionary of attributes and their values

#### **refresh(\*\*kwargs)**

Use this to make the device resource be represented by self.

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

This method makes an HTTP GET query against the device service. This method is run for its side-effects on self. If successful the instance attribute `__dict__` is replaced with the dict representing the device state. To figure out what that state is, run a subsequent query of the object like this: As with all CURDLE methods use a “requests\_params” dict to pass parameters to `requests.session.HTTPMETHOD`. See `test_requests_params.py` for an example. 

```
>>> resource_obj.refresh() >>> print(resource_obj.raw)
```

#### `update(**kwargs)`

Update the configuration of the resource on the BIG-IP®.

This method uses HTTP PUT alter the resource state on the BIG-IP®.

The attributes of the instance will be packaged as a dictionary. That dictionary will be updated with kwargs. It is then submitted as JSON to the device.

Various edge cases are handled: \* read-only attributes that are unchangeable are removed

#### Args:

**kwargs (dict):** Arbitrary number of keyword arguments.

Keys and associated values to alter on the device.

If kwargs has a `requests_params` key the corresponding dict will be passed to the underlying `requests.session.put` method where it will be handled according to that API.

## clientssl\_ciphers

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

BIG-IP® utility module

### REST URI

```
http://localhost/mgmt/tm/util/clientssl-ciphers
```

### GUI Path

N/A

### REST Kind

```
tm:util:clientssl-ciphers:*
```

*class*

**f5.bigip.tm.util.clientssl\_ciphers.Clientssl\_Ciphers**(*util*)  
[\[source\]](#)

Bases: `f5.bigip.resource.UnnamedResource`,  
`f5.bigip.mixins.CommandExecutionMixin`

BIG-IP® utility command

#### 📌 Note

This is an unnamed resource so it has no ~Partition~Name pattern at the end of its URI.

**create**(\*\*kwargs)

Create is not supported for unnamed resources

**Raises:**

UnsupportedMethod: If the method is used

**delete**(\*\*kwargs)

Delete is not supported for unnamed resources

**Raises:**

UnsupportedMethod: If the method is used

**exec\_cmd**(command, \*\*kwargs)

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

Wrapper method that can be changed in the inheriting classes.

#### **load(\*\*kwargs)**

Load is not supported for command execution

**Raises:**     UnsupportedOperation

#### **modify(\*\*patch)**

Modify the configuration of the resource on device based on patch

#### **raw**

Display the attributes that the current object has and their values.

**Returns:**     A dictionary of attributes and their values

#### **refresh(\*\*kwargs)**

Use this to make the device resource be represented by self.

This method makes an HTTP GET query against the device service. This method is run for its side-effects on self. If successful the instance attribute `__dict__` is replaced with the dict representing the device state. To figure out what that state is, run a subsequent query of the object like this: As with all CURDLE methods use a “requests\_params” dict to pass parameters to `requests.session.HTTPMETHOD`. See `test_requests_params.py` for an example. >>> `resource_obj.refresh()` >>> `print(resource_obj.raw)`

#### **update(\*\*kwargs)**

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

Update the configuration of the resource on the BIG-IP®.

This method uses HTTP PUT alter the resource state on the BIG-IP®.

The attributes of the instance will be packaged as a dictionary. That dictionary will be updated with kwargs. It is then submitted as JSON to the device.

Various edge cases are handled: \* read-only attributes that are unchangeable are removed

#### Args:

**kwargs (dict):** Arbitrary number of keyword arguments.

Keys and associated values to alter on the device.

If kwargs has a `requests_params` key the corresponding dict will be passed to the underlying `requests.session.put` method where it will be handled according to that API.

## dig

BIG-IP® utility module

### REST URI

`http://localhost/mgmt/tm/util/dig`

### GUI Path

N/A

### REST Kind

`tm:util:dig:*`

```
class f5.bigip.tm.util.dig.Dig(util) \[source\]
```

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

Bases: `f5.bigip.resource.UnnamedResource`,  
`f5.bigip.mixins.CommandExecutionMixin`

BIG-IP® utility command

### 📌 Note

This is an unnamed resource so it has no ~Partition~Name pattern at the end of its URI.

### `create(**kwargs)`

Create is not supported for unnamed resources

#### Raises:

UnsupportedMethod: If the method is used

### `delete(**kwargs)`

Delete is not supported for unnamed resources

#### Raises:

UnsupportedMethod: If the method is used

### `exec_cmd(command, **kwargs)`

Wrapper method that can be changed in the inheriting classes.

### `load(**kwargs)`

Load is not supported for command execution

**Raises:**     UnsupportedOperation

### `modify(**patch)`

Modify the configuration of the resource on device based on patch



## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

#### raw

Display the attributes that the current object has and their values.

**Returns:** A dictionary of attributes and their values

#### refresh(\*\*kwargs)

Use this to make the device resource be represented by self.

This method makes an HTTP GET query against the device service. This method is run for its side-effects on self. If successful the instance attribute `__dict__` is replaced with the dict representing the device state. To figure out what that state is, run a subsequent query of the object like this: As with all CURDLE methods use a “requests\_params” dict to pass parameters to `requests.session.HTTPMETHOD`. See `test_requests_params.py` for an example. 

```
>>> resource_obj.refresh() >>> print(resource_obj.raw)
```

#### update(\*\*kwargs)

Update the configuration of the resource on the BIG-IP®.

This method uses HTTP PUT alter the resource state on the BIG-IP®.

The attributes of the instance will be packaged as a dictionary. That dictionary will be updated with kwargs. It is then submitted as JSON to the device.

Various edge cases are handled: \* read-only attributes that are unchangeable are removed

#### Args:

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

**kwargs (dict):** Arbitrary number of keyword arguments.

Keys and associated values to alter on the device.

If kwargs has a `requests_params` key the corresponding dict will be passed to the underlying `requests.session.put` method where it will be handled according to that API.

## get\_dossier

BIG-IP® utility module

### REST URI

`http://localhost/mgmt/tm/util/get-dossier`

### GUI Path

N/A

### REST Kind

`tm:util:get-dossier:*`

```
class f5.bigip.tm.util.get_dossier.Get_Dossier(util)
[source]
```

Bases: `f5.bigip.resource.UnnamedResource`,  
`f5.bigip.mixins.CommandExecutionMixin`

BIG-IP® utility command

### ⓘ Note

This is an unnamed resource so it has not ~Partition~Name pattern at the end of its URI.

**create(\*\*kwargs)**

Create is not supported for unnamed resources

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

#### Raises:

UnsupportedMethod: If the method is used

#### `delete(**kwargs)`

Delete is not supported for unnamed resources

#### Raises:

UnsupportedMethod: If the method is used

#### `exec_cmd(command, **kwargs)`

Wrapper method that can be changed in the inheriting classes.

#### `load(**kwargs)`

Load is not supported for command execution

**Raises:**      `UnsupportedOperation`

#### `modify(**patch)`

Modify the configuration of the resource on device based on patch

#### `raw`

Display the attributes that the current object has and their values.

**Returns:**      A dictionary of attributes and their values

#### `refresh(**kwargs)`

Use this to make the device resource be represented by self.

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📄 F5 SDK API Docs

#### 📄 f5 package

##### 📄 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📄 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

This method makes an HTTP GET query against the device service. This method is run for its side-effects on self. If successful the instance attribute `__dict__` is replaced with the dict representing the device state. To figure out what that state is, run a subsequent query of the object like this: As with all CURDLE methods use a “requests\_params” dict to pass parameters to `requests.session.HTTPMETHOD`. See `test_requests_params.py` for an example. 

```
>>> resource_obj.refresh() >>> print(resource_obj.raw)
```

#### `update(**kwargs)`

Update the configuration of the resource on the BIG-IP®.

This method uses HTTP PUT alter the resource state on the BIG-IP®.

The attributes of the instance will be packaged as a dictionary. That dictionary will be updated with kwargs. It is then submitted as JSON to the device.

Various edge cases are handled: \* read-only attributes that are unchangeable are removed

#### Args:

**kwargs (dict):** Arbitrary number of keyword arguments.

Keys and associated values to alter on the device.

If kwargs has a `requests_params` key the corresponding dict will be passed to the underlying `requests.session.put` method where it will be handled according to that API.

**qkview**

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

BIG-IP® utility module

### REST URI

```
http://localhost/mgmt/tm/util/qkview
```

### GUI Path

N/A

### REST Kind

```
tm:util:qkview:*
```

```
class f5.bigip.tm.util.qkview.Qkview(util) \[source\]
```

Bases: `f5.bigip.resource.UnnamedResource`,  
`f5.bigip.mixins.CommandExecutionMixin`

BIG-IP® utility command

#### ! Note

This is an unnamed resource so it has no ~Partition~Name pattern at the end of its URI.

```
create(**kwargs)
```

Create is not supported for unnamed resources

Raises:

UnsupportedMethod: If the method is used

```
delete(**kwargs)
```

Delete is not supported for unnamed resources

Raises:

UnsupportedMethod: If the method is used

```
exec_cmd(command, **kwargs)
```

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

Wrapper method that can be changed in the inheriting classes.

#### **load(\*\*kwargs)**

Load is not supported for command execution

**Raises:**      `UnsupportedOperation`

#### **modify(\*\*patch)**

Modify the configuration of the resource on device based on patch

#### **raw**

Display the attributes that the current object has and their values.

**Returns:**      A dictionary of attributes and their values

#### **refresh(\*\*kwargs)**

Use this to make the device resource be represented by self.

This method makes an HTTP GET query against the device service. This method is run for its side-effects on self. If successful the instance attribute `__dict__` is replaced with the dict representing the device state. To figure out what that state is, run a subsequent query of the object like this: As with all CURDLE methods use a “requests\_params” dict to pass parameters to `requests.session.HTTPMETHOD`. See `test_requests_params.py` for an example. >>> `resource_obj.refresh()` >>> `print(resource_obj.raw)`

#### **update(\*\*kwargs)**

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

Update the configuration of the resource on the BIG-IP®.

This method uses HTTP PUT alter the resource state on the BIG-IP®.

The attributes of the instance will be packaged as a dictionary. That dictionary will be updated with kwargs. It is then submitted as JSON to the device.

Various edge cases are handled: \* read-only attributes that are unchangeable are removed

#### Args:

kwargs (dict): Arbitrary number of keyword arguments.

Keys and associated values to alter on the device.

If kwargs has a `requests_params` key the corresponding dict will be passed to the underlying `requests.session.put` method where it will be handled according to that API.

## serverssl\_ciphers

BIG-IP® utility module

### REST URI

`http://localhost/mgmt/tm/util/serverssl-ciphers`

### GUI Path

N/A

### REST Kind

`tm:util:serverssl-ciphers:*`

*class*

`f5.bigip.tm.util.serverssl_ciphers.Serverssl_Ciphers`(*util*)

[\[source\]](#)

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

Bases: `f5.bigip.resource.UnnamedResource`,  
`f5.bigip.mixins.CommandExecutionMixin`

BIG-IP® utility command

### 📌 Note

This is an unnamed resource so it has no ~Partition~Name pattern at the end of its URI.

#### `create(**kwargs)`

Create is not supported for unnamed resources

#### Raises:

UnsupportedMethod: If the method is used

#### `delete(**kwargs)`

Delete is not supported for unnamed resources

#### Raises:

UnsupportedMethod: If the method is used

#### `exec_cmd(command, **kwargs)`

Wrapper method that can be changed in the inheriting classes.

#### `load(**kwargs)`

Load is not supported for command execution

Raises: `UnsupportedOperation`

#### `modify(**patch)`

Modify the configuration of the resource on device based on patch



## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

#### raw

Display the attributes that the current object has and their values.

**Returns:** A dictionary of attributes and their values

#### refresh(\*\*kwargs)

Use this to make the device resource be represented by self.

This method makes an HTTP GET query against the device service. This method is run for its side-effects on self. If successful the instance attribute `__dict__` is replaced with the dict representing the device state. To figure out what that state is, run a subsequent query of the object like this: As with all CURDLE methods use a “requests\_params” dict to pass parameters to `requests.session.HTTPMETHOD`. See `test_requests_params.py` for an example. 

```
>>> resource_obj.refresh() >>> print(resource_obj.raw)
```

#### update(\*\*kwargs)

Update the configuration of the resource on the BIG-IP®.

This method uses HTTP PUT alter the resource state on the BIG-IP®.

The attributes of the instance will be packaged as a dictionary. That dictionary will be updated with kwargs. It is then submitted as JSON to the device.

Various edge cases are handled: \* read-only attributes that are unchangeable are removed

#### Args:

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

**kwargs** (dict): Arbitrary number of keyword arguments.

Keys and associated values to alter on the device.

If kwargs has a `requests_params` key the corresponding dict will be passed to the underlying `requests.session.put` method where it will be handled according to that API.

## unix\_ls

BIG-IP® utility module

### REST URI

`http://localhost/mgmt/tm/util/unix-ls`

### GUI Path

N/A

### REST Kind

`tm:util:unix-ls:*`

```
class f5.bigip.tm.util.unix_ls.Unix_Ls(util) \[source\]
```

Bases: `f5.bigip.resource.UnnamedResource`,  
`f5.bigip.mixins.CommandExecutionMixin`

**create**(\*\*kwargs)

Create is not supported for unnamed resources

**Raises:**

UnsupportedMethod: If the method is used

**delete**(\*\*kwargs)

Delete is not supported for unnamed resources

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

#### Raises:

UnsupportedMethod: If the method is used

#### **exec\_cmd**(*command*, *\*\*kwargs*)

Wrapper method that can be changed in the inheriting classes.

#### **load**(*\*\*kwargs*)

Load is not supported for command execution

**Raises:**     UnsupportedOperation

#### **modify**(*\*\*patch*)

Modify the configuration of the resource on device based on patch

#### **raw**

Display the attributes that the current object has and their values.

**Returns:**     A dictionary of attributes and their values

#### **refresh**(*\*\*kwargs*)

Use this to make the device resource be represented by self.

This method makes an HTTP GET query against the device service. This method is run for its side-effects on self. If successful the instance attribute `__dict__` is replaced with the dict representing the device state. To figure out what that state is, run a subsequent query of the object like this: As with all CURDLE methods use a “requests\_params” dict to pass parameters to

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

requests.session.HTTPMETHOD. See test\_requests\_params.py for an example. >>> resource\_obj.refresh() >>> print(resource\_obj.raw)

#### update(\*\*kwargs)

Update the configuration of the resource on the BIG-IP®.

This method uses HTTP PUT alter the resource state on the BIG-IP®.

The attributes of the instance will be packaged as a dictionary. That dictionary will be updated with kwargs. It is then submitted as JSON to the device.

Various edge cases are handled: \* read-only attributes that are unchangeable are removed

#### Args:

kwargs (dict): Arbitrary number of keyword arguments.

Keys and associated values to alter on the device.

If kwargs has a `requests_params` key the corresponding dict will be passed to the underlying `requests.session.put` method where it will be handled according to that API.

## unix\_mv

BIG-IP® utility module

### REST URI

`http://localhost/mgmt/tm/util/unix-mv`

### GUI Path

N/A

### REST Kind

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

```
tm:util:unix-mv:*
```

```
class f5.bigip.tm.util.unix_mv.Unix_Mv(util) \[source\]
```

Bases: `f5.bigip.resource.UnnamedResource`,  
`f5.bigip.mixins.CommandExecutionMixin`

**create(\*\*kwargs)**

Create is not supported for unnamed resources

**Raises:**

UnsupportedMethod: If the method is used

**delete(\*\*kwargs)**

Delete is not supported for unnamed resources

**Raises:**

UnsupportedMethod: If the method is used

**exec\_cmd(command, \*\*kwargs)**

Wrapper method that can be changed in the inheriting classes.

**load(\*\*kwargs)**

Load is not supported for command execution

**Raises:** `UnsupportedOperation`

**modify(\*\*patch)**

Modify the configuration of the resource on device based on patch

**raw**

Display the attributes that the current object has and their values.

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

**Returns:** A dictionary of attributes and their values

#### **refresh(\*\*kwargs)**

Use this to make the device resource be represented by self.

This method makes an HTTP GET query against the device service. This method is run for its side-effects on self. If successful the instance attribute `__dict__` is replaced with the dict representing the device state. To figure out what that state is, run a subsequent query of the object like this: As with all CURDLE methods use a “requests\_params” dict to pass parameters to `requests.session.HTTPMETHOD`. See `test_requests_params.py` for an example. 

```
>>> resource_obj.refresh() >>> print(resource_obj.raw)
```

#### **update(\*\*kwargs)**

Update the configuration of the resource on the BIG-IP®.

This method uses HTTP PUT alter the resource state on the BIG-IP®.

The attributes of the instance will be packaged as a dictionary. That dictionary will be updated with kwargs. It is then submitted as JSON to the device.

Various edge cases are handled: \* read-only attributes that are unchangeable are removed

#### **Args:**

**kwargs (dict):** Arbitrary number of keyword arguments.

Keys and associated values to alter on the device.

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📄 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

If kwargs has a `requests_params` key the corresponding dict will be passed to the underlying `requests.session.put` method where it will be handled according to that API.

## unix\_rm

BIG-IP® utility module

### REST URI

```
http://localhost/mgmt/tm/util/unix-rm
```

### GUI Path

N/A

### REST Kind

```
tm:util:unix-rm:*
```

```
class f5.bigip.tm.util.unix_rm.Unix_Rm(util) \[source\]
```

Bases: `f5.bigip.resource.UnnamedResource`,  
`f5.bigip.mixins.CommandExecutionMixin`

**create(\*\*kwargs)**

Create is not supported for unnamed resources

**Raises:**

UnsupportedMethod: If the method is used

**delete(\*\*kwargs)**

Delete is not supported for unnamed resources

**Raises:**

UnsupportedMethod: If the method is used

**exec\_cmd(command, \*\*kwargs)**

## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

Wrapper method that can be changed in the inheriting classes.

#### **load(\*\*kwargs)**

Load is not supported for command execution

**Raises:**     UnsupportedOperation

#### **modify(\*\*patch)**

Modify the configuration of the resource on device based on patch

#### **raw**

Display the attributes that the current object has and their values.

**Returns:**     A dictionary of attributes and their values

#### **refresh(\*\*kwargs)**

Use this to make the device resource be represented by self.

This method makes an HTTP GET query against the device service. This method is run for its side-effects on self. If successful the instance attribute `__dict__` is replaced with the dict representing the device state. To figure out what that state is, run a subsequent query of the object like this: As with all CURDLE methods use a “requests\_params” dict to pass parameters to `requests.session.HTTPMETHOD`. See `test_requests_params.py` for an example. >>> `resource_obj.refresh()` >>> `print(resource_obj.raw)`

#### **update(\*\*kwargs)**



## 🏠 F5 Python SDK

latest

User Guide

Developer Guide

### 📁 F5 SDK API Docs

#### 📁 f5 package

##### 📁 f5.bigip

f5.bigip module

resource module

mixins module

f5.bigip.cm

##### 📁 f5.bigip.tm

Subpackage List

Subpackages

f5.bigip.shared

f5.multi\_device

f5.sdk\_exception

Update the configuration of the resource on the BIG-IP®.

This method uses HTTP PUT alter the resource state on the BIG-IP®.

The attributes of the instance will be packaged as a dictionary. That dictionary will be updated with kwargs. It is then submitted as JSON to the device.

Various edge cases are handled: \* read-only attributes that are unchangeable are removed

#### Args:

**kwargs (dict):** Arbitrary number of keyword arguments.

Keys and associated values to alter on the device.

If kwargs has a `requests_params` key the corresponding dict will be passed to the underlying `requests.session.put` method where it will be handled according to that API.

← Previous

Next →

© Copyright 2017, F5 Networks Revision c831fe4b.

Built with [Sphinx](#) using a [theme](#) provided by [Read the Docs](#).