# JUNOS AUTOMATION AND REST API

Umberto Manferdini

# REST EVERYWHERE

- Many systems have REST APIs : JUNOS, Junos Space, Cloud Analytics Engine, Openclos, Contrail, Openstack, NSX …
- You first need to have the REST API documentation for your system.
- Then you can use a graphical REST Client (browser add-on: REST Easy, RESTClient, Postman) to start playing with REST APIs and learn more about REST APIs.
  - Graphical REST clients are for humans.
  - If you need automation and programmatic access, you have to use a command line REST client.
- You can then use Python as a REST Client to handle REST Calls. It is easy to parse the REST servers answers if they use a json format (json format is a dictionary).


Copyright © 2009 Juniper Networks, Inc. www.juniper.net

# JUNOS & REST

- JUNOS 15.1 supports REST API to submit RPCs
  - You can only read the database
  - The other 3 basic database operations (create-update-delete) are not supported
  - You can use HTTP get and post methods to submit RPCs to the REST Server.
  - You can retrieve data in XML or JSON
- The documentation is here:
  - https://www.juniper.net/documentation/en_US/junos15.1/information-products/pathway-pages/rest-api/rest-api.pdf
- REST configuration is under “system services” (default port is 3000)
- REST Explorer is an optional tool (GUI) for testing
- JUNOS CLI ouput with “| display json” is also available


Copyright © 2009 Juniper Networks, Inc. www.juniper.net

# REST IS A SERVICE

- Minimal configuration needed
- REST service enabled
- Optionally you can specify the port
- Optionally you can enable the REST API explorer for testing purposes

```
set system services rest http
set system services rest http rest-explorer
```


Copyright © 2009 Juniper Networks, Inc.

# ONBOARD REST EXPLORER

## REST-API explorer

JUNIPER NETWORKS

Single RPC  Multiple RPCs

HTTP method: GET

Required output format: JSON

RPC URL: /rpc/get-software-information

Username: pytraining

Password: ••••••••••

Submit

### Request Headers

```
GET /rpc/get-software-information HTTP/1.1
Authorization: Basic cHl0cmFpbmluZzpQb2NsYWIxMjM=
Accept: application/json
Content-Type: application/xml
```

### cURL request

```
curl http://172.30.177.170:3000/rpc/get-software-
information -u "pytraining:Poclab123" -H "Content-Type:
application/xml" -H "Accept: application/json"
```

### Response Headers

```
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
Date: Fri, 13 Nov 2015 11:04:01 GMT
Server: lighttpd/1.4.32
```

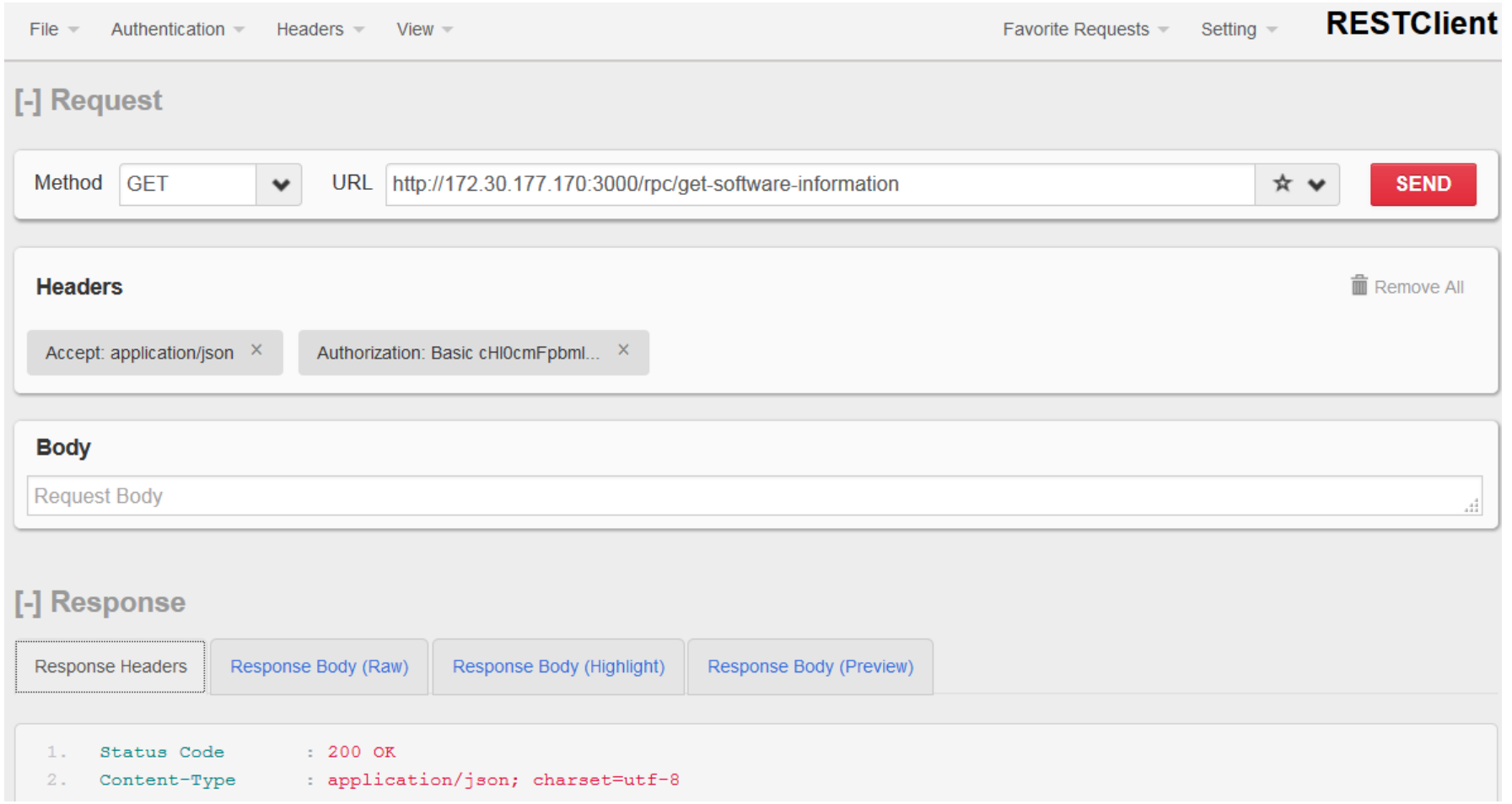
### Response Body

```
{
 "software-information" : [
 {
    "host-name" : [
    {
        "data" : "mx80-17"
    }
    ],
    "product-model" : [
    {
        "data" : "mx80-48t"
    }
    ],
    "product-name" : [
    {
        "data" : "mx80-48t"
    }
    ],
    "junos-version" : [
    {
        "data" : "15.1R2.9"
    }
    ],
```


Copyright © 2009 Juniper Networks, Inc.


JUNIPER NETWORKS

# TEST REST VIA FIREFOX





Copyright © 2009 Juniper Networks, Inc. www.juniper.net

# API & CURL

- CLI can be used to query devices
- We can use a common tool like CURL

```
pytraining@py-automation-master:~$ curl
http://172.30.177.170:3000/rpc/get-software-information
-u "pytraining:Poclab123"
-H "Content-Type: application/xml"
-H "Accept: application/json“
```


Copyright © 2009 Juniper Networks, Inc. www.juniper.net

# PYTHON TO QUERY THE WEB

```
>>> import requests

>>> r=requests.get("http://www.juniper.net")

>>> r.status_code
200

>>> print r.url
http://www.juniper.net/us/en/

>>> r.headers['Content-type']
'text/html;charset=UTF-8‘

>>> "NETWORK AUTOMATION" in r.content
True
```

- Python can talk with the WWW
- Need the requests module
- Can run a get method
- Page content saved into the “content” variable


Copyright © 2009 Juniper Networks, Inc. www.juniper.net

# PYTHON & REST API

```
>>> import requests

>>> from requests.auth import HTTPBasicAuth

r=requests.get('http://172.30.177.170:3000/rpc
/get-software-information',
auth=HTTPBasicAuth('pytraining', 'Poclab123'))

>>> r.status_code
200

>>> r.headers['Content-type']
'application/xml; charset=utf-8‘
```

- We can contact the device over the REST API port (3000)
- We need an authentication module as credentials are needed to query the device


Copyright © 2009 Juniper Networks, Inc. www.juniper.net

# REST API EXAMPLE

```
>>> my_headers = { 'Accept': 'application/json' }
>>> r = requests.get('http://172.30.177.170:3000/rpc/get-software-information',
auth=HTTPBasicAuth('pytraining', 'Poclab123'),headers=my_headers)

>>> type(r.json())
<type 'dict'>

>>> print r.json()['software-information'][0]['product-name'][0]['data']
mx80-48t

>>> print r.json()['software-information'][0]['host-name'][0]['data']
mx80-17

>>> print r.json()['software-information'][0]['junos-version'][0]['data']
15.1R2.9

>>> '15.1R2.9' in r.content
True
```


Copyright © 2009 Juniper Networks, Inc. www.juniper.net

everywhere