

# Utilisation de restconf sur le c800v

Netconf RESTCONF fournit un sous-ensemble simplifié de fonctionnalités NETCONF.

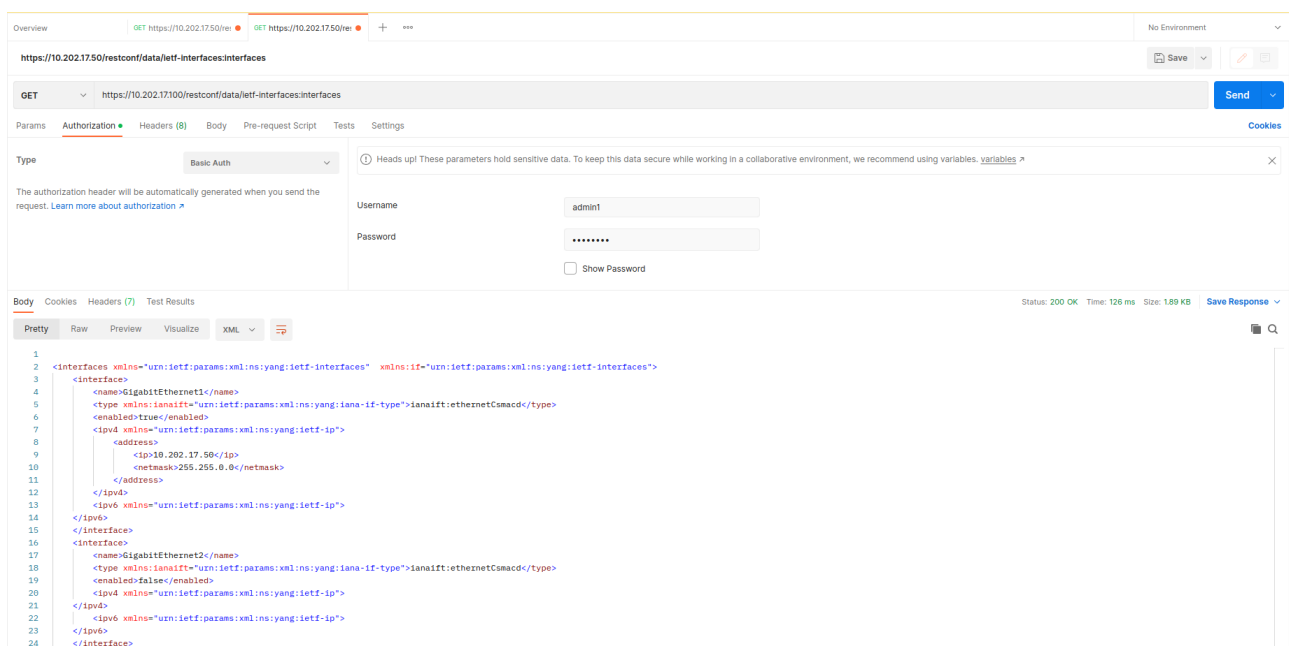
## Coté router

Activation avec la commande "restconf" dans le mode configuration terminal du router. et les commande suivante pour activer le serveur https et le mode d'authentification : "ip http secure-server", "ip http authentication local".

Pour obtenir verifier que tout est en etat de fonctionnement il existe la commande "show platform software yang-management process".

## Version graphique

On peut utiliser les différentes méthode en version graphique avec des utilitaires comme postman. Pour cela on doit désactiver les certificats SSL, choisir la méthode à utiliser, donner la requête souhaitée et quelques autres étapes ce qui nous permet au final d'obtenir, une fois envoyé, un résultat.



## Coté client

Coté du client (administrateur), on va utiliser du python pour réaliser notre administration.

## Utilisation GET

Ici on va utiliser la méthode GET, pour obtenir la configuration de nos différentes interfaces réseaux.

```
import json
```

```
import requests
requests.packages.urllib3.disable_warnings()

##### GET #####

api_url="https://10.202.17.200/restconf/data/ietf-
interfaces:interfaces" #requête que l'on souhaite effectuer

headers={"Accept":"application/yang-data+json","Content-
type":"application/yang-data+json"} # entête de la requête

basicauth=("admin1", "Root123#") #credentier de notre router

resp = requests.get(api_url, auth=basicauth, headers=headers,
verify=False) #requête "complete"

print(resp) # affichage du code reponse

response_json = resp.json()

print(json.dumps(response_json, indent=4)) # affichage de la reponse
sous forme de json
```

Retour de la requête precedente :

```
<Response [200]>
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "GigabitEthernet1",
        "type": "iana-if-type:ethernetCsmacd",
        "enabled": true,
        "ietf-ip:ipv4": {
          "address": [
            {
              "ip": "10.20.0.2",
              "netmask": "255.255.255.0"
            }
          ]
        },
        "ietf-ip:ipv6": {}
      },
      {
        "name": "GigabitEthernet2",
        "type": "iana-if-type:ethernetCsmacd",
        "enabled": true,
        "ietf-ip:ipv4": {
          "address": [
            {
```

```

        "ip": "10.202.17.200",
        "netmask": "255.255.0.0"
    }
    ],
    },
    "ietf-ip:ipv6": {}
},
{
    "name": "GigabitEthernet4",
    "type": "iana-if-type:ethernetCsmacd",
    "enabled": true,
    "ietf-ip:ipv4": {
        "address": [
            {
                "ip": "192.168.2.1",
                "netmask": "255.255.255.0"
            }
        ]
    },
    "ietf-ip:ipv6": {}
},
{
    "name": "Loopback1",
    "description": "loopback",
    "type": "iana-if-type:softwareLoopback",
    "enabled": true,
    "ietf-ip:ipv4": {
        "address": [
            {
                "ip": "1.1.1.1",
                "netmask": "255.255.255.255"
            }
        ]
    },
    "ietf-ip:ipv6": {}
}
]
}
}

```

## Ajout d'une loopback avec la methode post

Ici on vas utiliser la methode GET, pour obtenir la configuration de nos diferentes interfaces resaux.

```

import json
import requests
requests.packages.urllib3.disable_warnings()

headers={"Accept":"application/yang-data+json", "Content-

```

```
type":"application/yang-data+json"} #requête que l'on souhaite
efectuer

basicauth=("admin1", "Root123#") #credentier de notre router

api_url2 = "https://10.202.17.200/restconf/data/ietf-
interfaces:interfaces/interface=Loopback2" # entête de la requête

#crop de notre requête :
YangConfig = {
  "ietf-interfaces:interface": {
    "name": "Loopback2",
    "description": "La loopback deux retour",
    "type": "iana-if-type:softwareLoopback",
    "enabled": True,
    "ietf-ip:ipv4": {
      "address": [
        {
          "ip": "5.5.5.5",
          "netmask":"255.255.255.255"
        }
      ]
    },
    "ietf-ip:ipv6": {}
  }
}

resp =
requests.put(api_url2,data=json.dumps(YangConfig),auth=basicauth,headers=headers,verify=False)

if(resp.status_code >= 200 and resp.status_code <= 299):
    print("STATUS OK: {}".format(resp.status_code)) #affichage du
status
else:
    print('Error. Code d\'état : {} \n Message d\'erreur : {}
'.format(resp.status_code, resp.json())) #affichage du status
```

Retour :

```
STATUS OK: 201
```

Coté routeur la configuration a bien été prise :

```
c8000v-math#sh running-config interface loopback 2
Building configuration...

Current configuration : 100 bytes
!
interface Loopback2
  description La loopback deux retour
  ip address 5.5.5.5 255.255.255.255
end
```