Contents

T-Nova Orchestrator's AuthN/Z mService: GateKeeper

GateKeeper is the T-Nova Identity/Authorization micro-service that allows simple actions such as user-registration/authentication and inter-services authorization based on tokens. The sample (illustrative) workflows can be found here: Gatekeeper-workflows.

API Development Tracker

#	URI	Headers	HTTP Type	Body	Purpose	Impl. Status
01	/	None	GET	None	Discovery API, list of supported URIs	OK
02	/admin/user/	X-Auth-Token	GET	None	List of registered users	OK
03	/admin/user/	X-Auth-Token	POST	Required	New user registration endpoint	OK
04	/admin/user/{user-id}	X-Auth-Token	GET	None	Details on the user	OK
05	/admin/user/{user-id}	X-Auth-Token	PUT	Required	Update an existing user	OK
06	/admin/user/{user-id}	X-Auth-Token	DELETE	None	Delete an existing user	Not Implemented
07	/auth/{user-id}	X-Auth-Password	GET	None	Authentication request, list of any valid token(s) are returned	OK
08	/token/	X-Auth-Uid, X-Auth-Password or X-Auth-Token	POST	None	Authenticates and then generate a new token.	OK
09	/token/{token-uuid}	X-Auth-Token	GET	None	Details of an existing token.	Not Implemented
10	/token/{token-uuid}	X-Auth-Token	DELETE	None	Revoke an existing token.	Not Implemented
11	/token/validate/{token-uuid}	X-Auth-Service-Key or X-Auth-Uid	GET	None	Validate an existing token. Request can be made by an user himself, or by a registered service.	OK

12	/admin/service/	X-Auth-Token	GET	None	List of registered services.	OK
13	/admin/service/	X-Auth-Token	POST	Required	Registers a new service.	OK
14	/admin/service/{service-id}	X-Auth-Token	PUT	Required	Modifies the service parameters.	Not Implemented
15	/admin/service/{service-id}	X-Auth-Token	DELETE	None	Deletes an exiting service.	Not Implemented
16	/admin/service/{service-id}	X-Auth-Token	GET	None	Returns details of an exiting service.	Not Implemented

HTTP Error Codes Summary

The status codes for the REST calls outlined above are shown here. General HTTP codes such as 500 means internal server error, 404 means resource not found, etc. **Anywhere, when admin token is expected, especially in the admin APIs, if token does not belong to a admin user, or invalid a HTTP 401 is returned.**

API #	URI	Return Code	Reason
01	/ [GET]	200	OK
02	/admin/user/ [GET]	200	OK
03	/admin/user/ [POST]	412	User already exists
03	/admin/user/ [POST]	200	User added successfully
04	/admin/user/{user-id} [GET]	200	User exists, details returned
04	/admin/user/{user-id} [GET]	404	User not found
05	/admin/user/{user-id}		
06	/admin/user/{user-id}		
07	/auth/{user-id} [GET}	202	Valid Password
07	/auth/{user-id} [GET]	401	Invalid Password
08	/token/ [POST]	200	Token generated successfully
09	/token/{token-uuid}		
10	/token/{token-uuid}		
11	/token/validate/{token-uuid} [GET]	200	Token validation successful.
11	/token/validate/{token-uuid} [GET]	406	Token validation failed.
12	/admin/service/ [GET]	200	List of services.
13	/admin/service/ [POST]	200	Service successfully registered.
13	/admin/service/ [POST]	400	Malformed request, incorrect POST data
13	/admin/service/ [POST]	412	Service already exists, request failed.

API Usage Details

01. / GET

Usage Example

```
curl -X GET auth.piyush-harsh.info:8000/ --header "Content-
Type:application/json"
```

Sample Response

```
[
        {
                "metadata":
                {
                        "source": "T-Nova-AuthZ-Service"
                },
                "info":
                [
                        {
                                "msg": "Welcome to the T-Nova-AuthZ-Service",
                                "purpose": "REST API Usage Guide",
                                "disclaimer": "It's not yet final!",
                                "notice": "Headers and body formats are not
defined here."
                        }
                ],
                "api":
                [
                        {
                                "uri": "/",
                                "method": "GET",
                                "purpose": "REST API Structure and Capability
Discovery"
                        },
                        {
                                "uri": "/admin/user/",
                                "method": "GET",
                                "purpose": "Admin API to get list of all
users"
                        },
                        {
                                "uri": "/admin/user/",
                                "method": "POST",
                                "purpose": "Create a new user"
                        },
                        {
                                "uri": "/admin/user/{user-id}",
                                "method": "GET",
                                "purpose": "Admin API to get detailed info of
a particular user"
                        },
                        {
                                "uri": "/admin/user/{user-id}",
                                "method": "PUT",
                                "purpose": "Admin API to modify details of a
particular user"
                        },
                        {
                                "uri": "/admin/user/{user-id}",
                                "method": "DELETE",
                                "purpose": "Admin API to delete a particular
user"
                        },
                        {
                                "uri": "/token/",
                                "method": "POST",
                                "purpose": "API to request a new service
token"
                        },
                        {
                                "uri": "/token/{token-uuid}",
                                "method": "GET",
                                "purpose": "Get details of this token,
lifetime, user or service id of the creator, etc."
                        },
                        {
                                "uri": "/token/{token-uuid}",
                                "method": "DELETE",
                                "purpose": "Revoke or essentially delete an
existing token."
                        },
                        {
                                "uri": "/token/validate/{token-uuid}",
                                "method": "GET",
                                "purpose": "Validate a existing token,
OK/Not-OK type status response"
                        },
                        {
                                "uri": "/auth/{user-id}",
                                "method": "GET",
                                "purpose": "Authenticate an existing user.
OK/Not-OK type response."
                        },
                        {
                                "uri": "/admin/service/",
                                "method": "GET",
                                "purpose": "Lists all registered services."
                        },
                        {
                                "uri": "/admin/service/",
                                "method": "POST",
                                "purpose": "Registers a new service with
gatekeeper."
                        }
                ]
```

```
        }
]
```

02 /admin/user/ GET

Usage Example

```
curl -X GET auth.piyush-harsh.info:8000/admin/user/ --header "Content-
Type:application/json" --header "X-Auth-Token:79E9EEF8-FC4B-43FD-
B1B2-4575D92864DC"
```

Sample Response

```
{
  "metadata": {
    "source": "T-Nova-AuthZ-Service"
  },
  "info": [
    {
      "msg": "list of active users"
    }
  ],
  "userlist": [
    "piyush"
  ]
}
```

03 /admin/user/ POST

IMPORTANT: Currently the user-capabilities or allowed-services-list is not utilized, but this will be modified very soon.

Usage Example

```
curl -X POST auth.piyush-harsh.info:8000/admin/user/ --header "Content-
Type:application/json" --header "X-Auth-Token:79E9EEF8-FC4B-43FD-
B1B2-4575D92864DC" -d
'{"username":"piyush","password":"somepass","isadmin":"n",
"accesslist":"ALL"}'
```

Sample Response (if created successfully)

```
{
        "metadata":
        {
                "source": "T-Nova-AuthZ-Service"
        },
        "info":
        [
                {
                        "msg": "user created successfully",
                        "auth-uri": "/auth/3",
                        "admin-uri": "/admin/user/3",
                        "id": "3"
                }
        ]
}
```

Sample Response (if in error)

```
{
        "metadata":
        {
                "source": "T-Nova-AuthZ-Service"
        },
        "info":
        [
                {
                        "msg": "User already exists."
                }
        ]
}
```

04 /admin/user/{user-id} GET

Usage Example

```
curl -X GET auth.piyush-harsh.info:8000/admin/user/2 --header "Content-
Type:application/json" --header "X-Auth-Token:79E9EEF8-FC4B-43FD-
B1B2-4575D92864DC"
```

Sample Response (if user exists)

```
{
        "metadata":
        {
                "source": "T-Nova-AuthZ-Service"
        },
        "info":
        [
                {
                        "msg": "Account details.",
                        "username": "piyush",
                        "isadmin": "n",
                        "capabilitylist": "udr,rc,billing"
                }
        ]
}
```

Sample Response (if user does not exist)

```
{
        "metadata":
        {
                "source": "T-Nova-AuthZ-Service"
        },
        "info":
        [
                {
                        "msg": "User not found."
                }
        ]
}
```

05 /admin/user/{user-id} PUT

Usage Examples

```
url -X PUT auth.piyush-harsh.info:8000/admin/user/2 --header "Content-
Type:application/json" --header "X-Auth-Token:79E9EEF8-FC4B-43FD-
B1B2-4575D92864DC" -d '{"isadmin":"y"}'
url -X PUT auth.piyush-harsh.info:8000/admin/user/2 --header "Content-
Type:application/json" --header "X-Auth-Token:79E9EEF8-FC4B-43FD-
B1B2-4575D92864DC" -d '{"accesslist":"servicex,servicey"}'
url -X PUT auth.piyush-harsh.info:8000/admin/user/2 --header "Content-
Type:application/json" --header "X-Auth-Token:79E9EEF8-FC4B-43FD-
B1B2-4575D92864DC" -d '{"accesslist":"servicex,servicey","isadmin":"y"}'
```

Sample Response (if update successful)

```
{
    "metadata":
    {
        "source": "T-Nova-AuthZ-Service"
    },
    "info":
    [
        {
            "msg": "Update Successful."
        }
    ]
}
```

Sample Response (if update failed)

```
{
        "metadata":
        {
                "source": "T-Nova-AuthZ-Service"
        },
        "info":
        [
                {
                        "msg": "Update Failed."
                }
        ]
}
```

07 /auth/{user-id} GET

Usage Example

```
curl -X GET auth.piyush-harsh.info:8000/auth/1 --header "Content-Type:application/json" --header "X-Auth-Password:somepass"
```

Sample Response (if successful)

```
{
    "metadata":
    {
        "source": "T-Nova-AuthZ-Service"
    },
    "info":
    [
        {
            "msg": "Authentication Successful."
        }
    ],
    "tokenlist":
    {
        "id":
                [
                        "3E055F4B-B10F-40FF-9FCD-C55E3E94062B"
                ],
        "valid-until":
                [
                        "2015-06-17 18:53:06 +0200 CEST"
                ]
    }
}
```

Sample Response (if unsuccessful)

```
{
        "metadata":
        {
                "source": "T-Nova-AuthZ-Service"
        },
        "info":
        [
                {
                        "msg": "Incorrect Password."
                }
        ]
}
```

08 /token/ POST

Usage Example

```
curl -X POST auth.piyush-harsh.info:8000/token/ --header "Content-
Type:application/json" --header "X-Auth-Password:somepass" --header "X-Aut-
-Uid:1"
```

Sample Response (if successful)

```
{
    "metadata":
    {
        "source": "T-Nova-AuthZ-Service"
    },
    "info":
    [
        {
            "msg": "Token Details."
        }
    ],
    "token":
    {
        "id":"13D707C4-BF77-4CE3-98D9-772D51A3BCC0",
        "valid-until":"2015-06-17 19:03:58.616306246 +0200 CEST"
    }
}
```

Sample Response (if unsuccessful)

```
{
        "metadata":
        {
                "source": "T-Nova-AuthZ-Service"
        },
        "info":
        [
                {
                        "msg": "Incorrect Password."
                }
        ]
}
```

11 /token/validate/{token-uuid} GET

Usage Example

```
curl -X GET auth.piyush-harsh.info:8000/token/validate/2D2C474C-9438-4-
79-B9AE-86BBB9E71C07 --header "Content-Type:application/json" --header "X-
Auth-Uid:1"
```

Sample Response (if successful)

```
{
   "metadata":
   {
       "source": "T-Nova-AuthZ-Service"
   },
   "info":
   [
        {
           "msg": "Validation Successful."
       }
   ]
}
```

Sample Response (if unsuccessful)

```
{
   "metadata":
   {
       "source": "T-Nova-AuthZ-Service"
   },
   "info":
   [
       {
           "msg": "Validation Failed."
       }
   ]
}
```

12 /admin/service/ GET

Usage Example

```
curl -X GET auth.piyush-harsh.info:8000/admin/service/ --header "Content-Type:application/json" --header "X-Auth-Token:79E9EEF8-FC4B-43FD-B1B2-4575D92864DC"
```

Sample Response (if successful)

```
{
    "metadata":
    {
        "source": "T-Nova-AuthZ-Service"
    },
    "info":
    [
        {
            "msg": "Registered Service List."
        }
    ],
    "servicelist":
    {
        "service-key":
                [
                        "07868B93-A853-4163-96E4-94FB879E8E97"
                ],
        "shortname":
                [
                        "vnfdpars"
                ]
    }
}
```

13 /admin/service/ POST

Usage Example

```
curl -X POST auth.piyush-harsh.info:8000/admin/service/ --header "Content-Type:application/json" --header "X-Auth-Token:79E9EEF8-FC4B-43FD-B1B2-4575D92864DC" -d '{"shortname":"vnfdpars","description":"this service parses the vnf descriptors"}'
```

IMP: shortname is limited to 16 characters only.

Sample Response (if successful)

```
{
        "metadata":
        {
                "source": "T-Nova-AuthZ-Service"
        },
        "info":
        [
                {
                        "msg": "service registered successfully",
                        "service-uri": "/service/2",
                        "service-key": "ADF46F8F-6B73-49A1-A345-0EEA669E0272"
                }
        ]
}
```

Sample Response (if unsuccessful)

```
{
        "metadata":
        {
                "source": "T-Nova-AuthZ-Service"
        },
        "info":
        [
                {
                        "msg": "Service with this shortname already exists."
                }
        ]
}
```