

HTTP API

TekRADIUS provides a JSON based HTTP REST API when the HTTP interface is enabled.

Types

OperationTypes

```
GetToken
ListActiveSessions
GetActiveSessionEntry
ClearActiveSessionEntry
KillActiveSessionEntry
DisconnectActiveSessionEntry
UpdateActiveSessionEntry
ListUserProfiles
GetUserProfileEntry
AddUserProfileEntry
DeleteUserProfileEntry
AddUserAttributeEntry
DeleteUserAttributeEntry
EnableUserProfile
DisableUserProfile
ListGroupProfiles
GetGroupProfileEntry
AddGroupProfileEntry
DeleteGroupProfileEntry
AddGroupAttributeEntry
DeleteGroupAttributeEntry
EnableGroupProfile
DisableGroupProfile
ListClientProfiles
GetClientProfileEntry
AddClientProfileEntry
DeleteClientProfileEntry
EnableClientProfileEntry
DisableClientProfileEntry
GetCounters
ResetCounters
GetLogs
ClearLogs
SendGASecret
```

AttrEvalTypes

```
Check
SuccessReply
FailureReply
Informational
DHCPSuccessReply
DHCPFailureReply
CoASet
CoAReset
```

Functions

Get Token [POST /gettoken]

Request

```
{
  "APIUsername": "kaplan",
  "APIPassword": "1234"
}
```

Response

```
{
  "Operation": "GetToken",
  "Success": true,
  "Response": "a080160c29296f816faa60429edf08ef"
}
```

Tokens are issued for Session Timeout duration specified in Settings / Parameters / HTTP Interface. Every operation extends its duration by one minute. API User must be a local TekRADIUS user profile with HTTP-Access-Level = Admin (Check) attribute.

List Active Sessions [POST /sessions]

Request

```
{
  "Token": "1234",
  "OperationType": "ListActiveSessions"
}
```

Response

```
[
  {"Username": "kaplan", "tracid": "DEFG"},
  {"Username": "dwadley", "tracid": "ABCD"}
]
```

The following attributes can be used for filtering: Acct-Session-Id, User-Name, Group-Name, NAS-IP-Address, NAS-Identifier, NAS-Port, NAS-Port-Type, NAS-Port-Id, Service-Type, Framed-IP-Address, Calling-Station-Id, Called-Station-Id.

Request	Response
<pre>{ "Token": "1234", "OperationType": "ListActiveSessions", "Data": { "attribute": "User-Name", "value": "kap" } }</pre>	<pre>[{ "Username": "kaplan", "tracid": "DEFG" }]</pre>

Get Active Session Entry [POST /sessions]

Request	Response
<pre>{ "Token": "1234", "OperationType": "GetActiveSessionEntry", "Data": { "tracid": ["ABCD"] } }</pre>	<pre>[{ "tracid": "ABCD", "SessionID": "8070F009", "TimeStamp": "/Date(1706462707187)/", "Username": "dwadley", "GroupName": "Default", : "NasPortId": "wlan-Hotspot", "ServiceType": "", "FramedIPAddr": "10.5.50.4", "CallingStationID": "AC:A9:37:B6:06:FF", "CalledStationID": "hotspot1" }]</pre>

Clear|Kill Active Session Entry [POST /sessions]

Request	Response
<pre>{ "Token": "1234", "OperationType": "ClearActiveSessionEntry KillActiveSessionEntry", "Data": { "tracid": ["ABCD"] } }</pre>	<pre>{ "Operation": "ClearActiveSessionEntry KillActiveSessionEntry", "Success": true, "Message": "" }</pre>

Disconnect|Update Active Session Entry [POST /sessions]

Request	Response
<pre>{ "Token": "1234", "OperationType": "DisconnectActiveSessionEntry UpdateActiveSessionEntry", "Data": { "tracid": ["ABCD"], "attributes": ["ericsson-ab 4=mycontext"] } }</pre>	<pre>{ "Operation": "DisconnectActiveSessionEntry UpdateActiveSessionEntry", "Success": true, "Message": "" }</pre>

List User Profiles [POST /users]

Request	Response
<pre>{ "Token": "1234", "OperationType": "ListUserProfiles" }</pre>	<pre>[{ "Username": "dwadley", "tracid": "ABCD" }]</pre>

You can add Data section with `selectedusers` field to filter user profiles.

Request

```
{
  "Token": "1234",
  "OperationType": "ListUserProfiles",
  "Data": {
    "selectedusers": "kaplan,dwadley"
  }
}
```

Response

```
{
  "Operation": "ClearActiveSessionEntry|KillActiveSessionEntry",
  "Success": true,
  "Message": ""
}
```

Enable User Profile(s) [POST /users]

Request

```
{
  "Token": "1234",
  "OperationType": "EnableUserProfile",
  "Data": {
    "selectedusers": "kaplan"
  }
}
```

Response

```
{
  "Operation": "EnableUserProfile",
  "Success": true,
  "Response": "User profile 'kaplan' is enabled."
}
```

Disable/Delete User Profile(s) [POST /users]

Request

```
{
  "Token": "1234",
  "OperationType": "DisableUserProfile|DeleteUserProfileEntry",
  "Data": {
    "selectedusers": "kaplan"
  }
}
```

Response

```
{
  "Operation": "DisableUserProfile|DeleteUserProfileEntry",
  "Success": true,
  "Response": "User profile 'kaplan' is disabled|deleted."
}
```

Add User Profile Entry [POST /users]

Request

```
{
  "Token": "1234",
  "OperationType": "AddUserProfileEntry",
  "Data": {
    "username": "myuser",
    "password": "mypassword",
    "group": "AlternativeGroup"
  }
}
```

Response

```
{
  "Operation": "AddUserProfileEntry",
  "Success": true,
  "Response": "User profile 'myuser' is added to group 'AlternativeGroup'."
}
```

Get User Profile Entry [POST /users]

Request

```
{
  "Token": "1234",
  "OperationType": "GetUserProfileEntry",
  "Data": {
    "selectedusers|username": "kaplan"
  }
}
```

Response

```
[
  {
    "Attribute": "Group-Name",
    "Type": "Check",
    "Value": "Blank"
  },
  {
    "Attribute": "User-Password",
    "Type": "Check",
    "Value": "*****"
  },
  {
    "Attribute": "NAS-Vendor",
    "Type": "Check",
    "Value": "ietf"
  }
]
```

Add User Attribute Entry [POST /users]

Request

```
{
  "Token": "1234",
  "OperationType": "AddUserAttributeEntry",
  "Data": {
    "selectedusers": "myuser",
    "attribute": "Calling-Station-Id",
    "attrtype": "Check",
    "value": "AA:BB:CC:11:22:33"
  }
}
```

Response

```
{
  "Operation": "AddUserAttributeEntry",
  "Success": true,
  "Response": "Attribute 'Calling-Station-Id' is added for the user profile 'myuser'."
}
```

Delete User Attribute Entry [POST /users]

Request

```
{
  "Token": "1234",
  "OperationType": "DeleteUserAttributeEntry",
  "Data": {
    "selectedusers": "myuser",
    "attribute": "Calling-Station-Id",
    "attrtype": "Check"
  }
}
```

Response

```
{
  "Operation": "DeleteUserAttributeEntry",
  "Success": true,
  "Response": "Calling-Station-Id (ietf|31) is deleted defined for the user profile 'myuser'."
}
```

List Group Profiles [POST /groups]

Request

```
{
  "Token": "1234",
  "OperationType": "ListGroupProfiles"
}
```

Response

```
[
  "2FA",
  "2FA-GA",
  "AlternativeGroup1",
  "AlternativeGroup2",
  "AlternativeGroup3",
  "Blank",
  "Default"
]
```

You can add Data section with `selectgroups` field to filter group profiles.

Enable Group Profile(s) [POST /groups]

Request

```
{
  "Token": "1234",
  "OperationType": "EnableGroupProfile",
  "Data": {
    "selectedgroups": "Blank"
  }
}
```

Response

```
{
  "Operation": "EnableGroupProfile",
  "Success": true,
  "Response": "Group profile 'Blank' is enabled."
}
```

Disable/Delete Group Profile(s) [POST /groups]

Request

```
{
  "Token": "1234",
  "OperationType": "DisableGroupProfile|DeleteGroupProfileEntry",
  "Data": {
    "selectedgroups": "Blank"
  }
}
```

Response

```
{
  "Operation": "DisableGroupProfile|DeleteGroupProfileEntry",
  "Success": true,
  "Response": "Group profile 'Blank' is disabled|deleted."
}
```

Add Group Profile Entry [POST /groups]**Request**

```
{
  "Token": "1234",
  "OperationType": "AddGroupProfileEntry",
  "Data": {
    "groupname": "MyGroup"
  }
}
```

Response

```
{
  "Operation": "AddGroupProfileEntry",
  "Success": true,
  "Response": "Group profile MyGroup is added."
}
```

Get Group Profile Entry [POST /groups]**Request**

```
{
  "Token": "1234",
  "OperationType": "GetGroupProfileEntry",
  "Data": {
    "selectedgroups|groupname": "TestGroup2"
  }
}
```

Response

```
[
  {
    "Attribute": "Group-Name",
    "Type": "Check",
    "Value": "Blank"
  },
  {
    "Attribute": "User-Password",
    "Type": "Check",
    "Value": "*****"
  },
  {
    "Attribute": "NAS-Vendor",
    "Type": "Check",
    "Value": "ietf"
  }
]
```

Add Group Attribute Entry [POST /groups]**Request**

```
{
  "Token": "1234",
  "OperationType": "AddGroupAttributeEntry",
  "Data": {
    "groupname": "MyGroup",
    "attribute": "Calling-Station-Id",
    "attrtype": "Check",
    "value": "AA:BB:CC:11:22:33"
  }
}
```

Response

```
{
  "Operation": "AddGroupAttributeEntry",
  "Success": true,
  "Response": "Attribute 'Calling-Station-Id' is added for the group 'MyGroup'."
}
```

Delete Group Attribute Entry [POST /groups]**Request**

```
{
  "Token": "1234",
  "OperationType": "DeleteGroupAttributeEntry",
  "Data": {
    "groupname": "MyGroup",
    "attribute": "Calling-Station-Id",
    "attrtype": "Check"
  }
}
```

Response

```
{
  "Operation": "DeleteGroupAttributeEntry",
  "Success": true,
  "Response": "Calling-Station-Id' is deleted defined for group profile 'MyGroup'."
}
```

List Client Profiles [POST /clients]**Request**

```
{
  "Token": "1234",
  "OperationType": "ListClientProfiles"
}
```

Response

```
[
  "10.142.192.171",
  "10.142.192.172",
  "126.10.10.1",
  "126.10.10.2",
  "192.168.1.51"
]
```

Get Client Profile Entry [POST /clients]**Request**

```
{
  "Token": "1234",
  "OperationType": "GetClientProfileEntry",
  "Data": {
    "NAS": "192.168.88.13"
  }
}
```

Response

```
[
  {
    "NAS": "192.168.88.13",
    "Secret": "test",
    "Vendor": "ietf",
    "UserRegEx": "",
    "KillCommand": "",
    "InterimUpdatePeriod": 0,
    "Group": "Default",
    "Description": "",
    "Enabled": "Yes"
  }
]
```

Add Client Profile Entry [POST /clients]**Request**

```
{
  "Token": "1234",
  "OperationType": "AddClientProfileEntry",
  "Data": {
    "NAS": "192.168.66.1",
    "Secret": "test",
    "Vendor": "ietf",
    "UserRegEx": "",
    "KillCommand": "",
    "InterimUpdatePeriod": 0,
    "Group": "Default",
    "Description": "",
    "Enabled": "Yes"
  }
}
```

Response

```
{
  "Operation": "AddClientProfileEntry",
  "Success": true,
  "Response": "Client entry 192.168.66.1 is updated (Enabled)."
}
```

Delete Client Profile Entry [POST /clients]**Request**

```
{
  "Token": "1234",
  "OperationType": "DeleteClientProfileEntry",
  "Data": {
    "NAS": "192.168.66.1"
  }
}
```

Response

```
{
  "Operation": "DeleteClientProfileEntry",
  "Success": false,
  "Response": "Client entry 192.168.66.1 is deleted."
}
```

Enable/Disable Client Profile Entry [POST /clients]**Request**

```
{
  "Token": "1234",
  "OperationType": "EnableClientProfileEntry|DisableClientProfileEntry",
  "Data": {
    "NAS": "192.168.88.13"
  }
}
```

Response

```
{
  "Operation": "EnableClientProfileEntry|DisableClientProfileEntry",
  "Success": false,
  "Response": "Client entry 192.168.66.1 is deleted."
}
```

Read Logs [POST /log]

Request	Response
<pre>{ "Token": "1234", "OperationType": "GetLogs" }</pre>	<pre>[{ "EntryDateTime": "/Date(1681727315000)/", "Type": "Information", "Message": "TekRADIUS Service is being stopped." }, { "EntryDateTime": "/Date(1681727315000)/", "Type": "Information", "Message": " TekRADIUS service is stopped." }]</pre>

Clear Logs [POST /log]

Request	Response
<pre>{ "Token": "1234", "OperationType": "GetCounters" }</pre>	<pre>{ "Operation": "GetCounters", "Success": true, "Message": "Application log is cleared" }</pre>

Read Counters [POST /counters]

Request	Response
<pre>{ "Token": "1234", "OperationType": "GetCounters" }</pre>	<pre>[{ "Name": "Number of Active Sessions", "Value": "1" }, : { "Name": "Number of RADIUS Accounting Timeouts", "Value": "0" }]</pre>

Clear Counters [POST /counters]

Request	Response
<pre>{ "Token": "1234", "OperationType": "ResetCounters" }</pre>	<pre>{ "Operation": "ResetCounters", "Success": true, "Message": "Counters are cleared" }</pre>

Operation type is case sensitive. You enable, disable and delete multiple user and group profiles by adding Data section with selectedgroups variable:

```
{
  "Token": "1234",
  "OperationType": "DisableGroupProfile ",
  "Data": {
    "selectedgroups": "Blank,AnotherGroup,MyGroup"
  }
}
```

Using HTTP API with TLS transport (*HTTPS*) is strongly recommended.