REST API User Guide

2019

STEALTHbits TECHNOLOGIES

200 Central Avenue | Hawthorne, NJ 07506
P. +1-201-447-9300 | F. +1-201-447-1818

Table of Contents

Web Service REST API for Applications Accessing Data Remotely

The StealthAUDIT REST API is integrated into the Web Service as an endpoint using an OAuth 2.0 client credentials grant for authentication and providing two types of access roles:

- → Full Privilege – Execute jobs and read data
- → Read-Only – Read data only

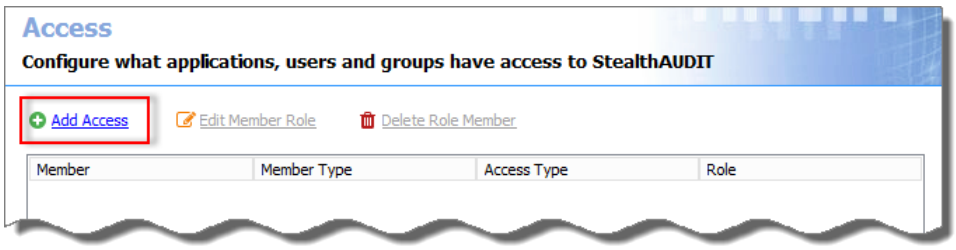
The client provides the access token in the HTTP header in the following format:

```
GET /api/v1/data/SA_ADInventory_UsersView/rows HTTP/1.1

Host: stealthaudit.company.com

Authorization: Bearer N4ahquT7rXuiEEeUiNfKD0TjUq7JB9DS
```

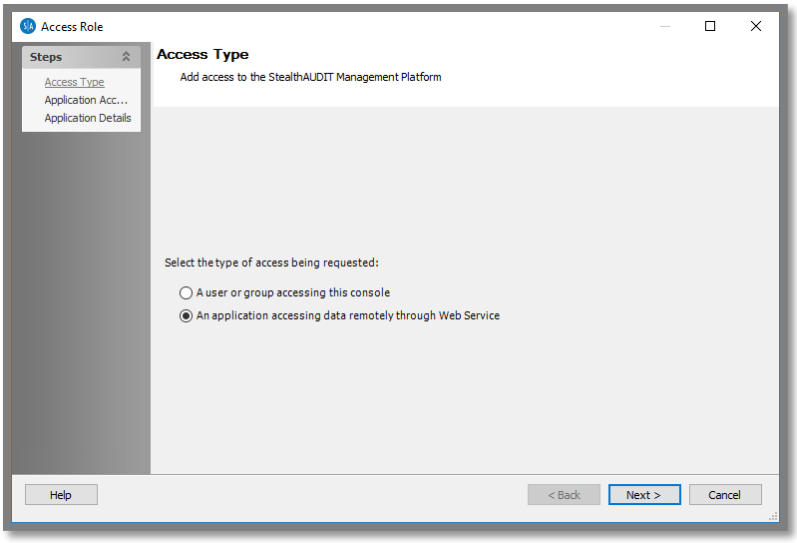
See the MDN Web Docs The general HTTP authentication framework article for additional information.

Assign Application Access through the Web Service

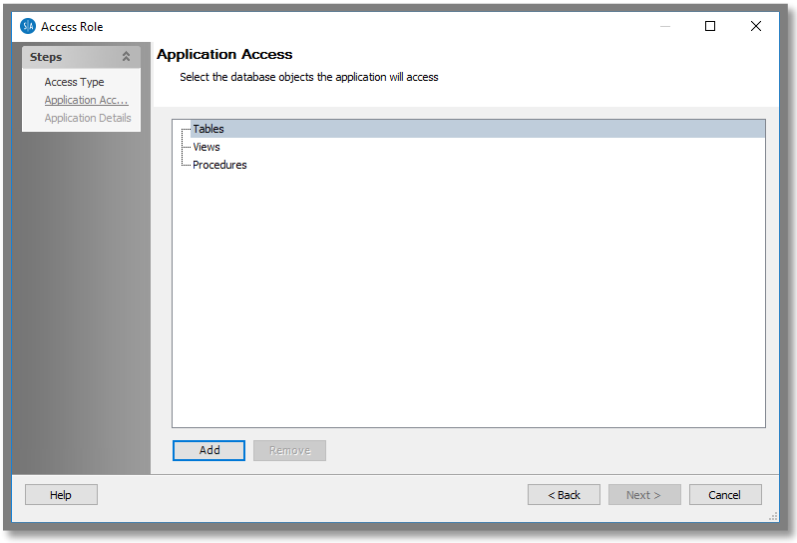
An application can be assigned to access data remotely through the Web Service. Follow the steps to assign roles in the StealthAUDIT Console.



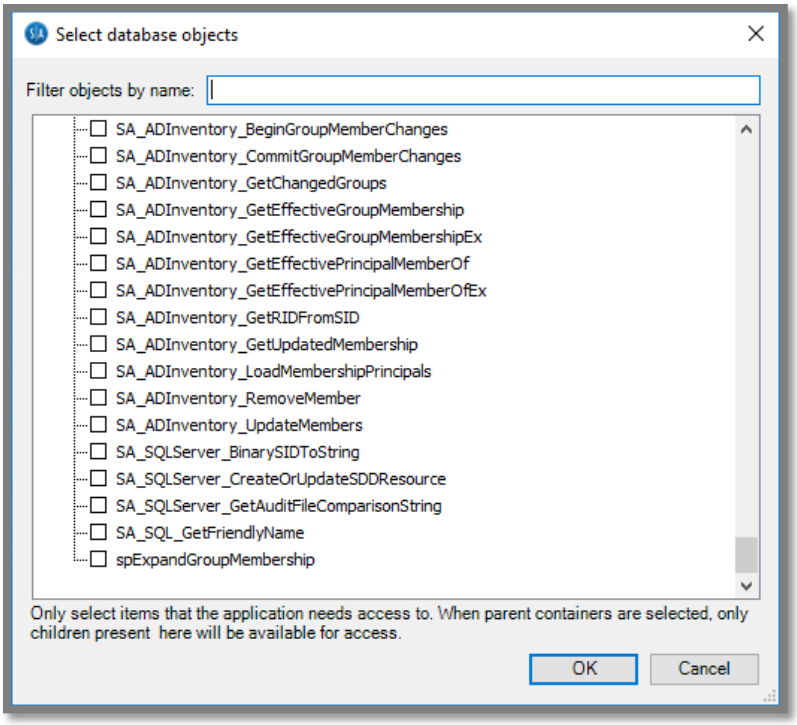
Step 1 – Click the **Add Access** link in the upper-right corner of the Roles Access view. The Access Type wizard opens.



Step 2 – Select the **An application accessing data remotely through Web Service** radio button. Click **Next**. The Application Access window opens.



Step 3 – The Application Access window displays a list of objects available in the database that are available for access. Select the database objects the application will access and click **Add** to open the Select database objects window.



Step 4 – Select the database objects to access and then click **OK** to return to the Application Access page.

NOTE: Only select items that the application needs to access. Type in the Filter objects by name box to filter the list of objects by the characters entered.

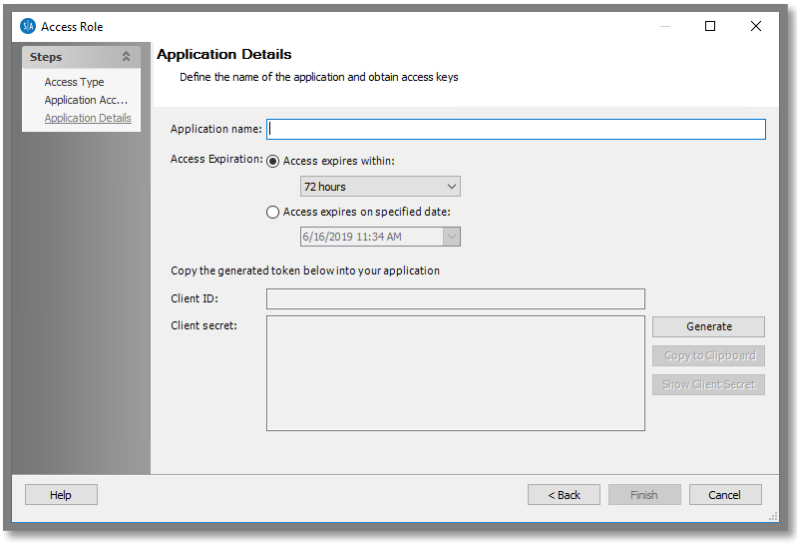
→ Selecting a parent node in the tree automatically selects all children in addition to the parent

→ Selecting a child automatically selects the parent

→ Unselecting a child when the parent is selected automatically puts the parent into an indeterminate state

→ Selecting any child puts the parent into an intermediate state

Click **Next.**



Step 5 – On the Application Details window, define the name of the application and generate the app token.

- → Application name – The name of the application accessing that data
- → Access Expiration – The expiration for the client secret. Select a radio button for the desired access expiration:
  - ☐ Access expires within – Select a timeframe from the drop-down list. The default is 72 hours.
  - ☐ Access expires on specified date – Select a date from the drop-down list
- → Generate – Click this button to generate the Client ID and Client secret
- → Client ID – Copy the Client ID into the application accessing data remotely through the Web Service
- → Client secret – Copy the Client secret into the application accessing data remotely through the Web Service

Step 6 – Click **Finish** to confirm the changes.

The next step is to use the Client ID and Client Secret to obtain an access token. This token is used to get data from the StealthAUDIT endpoint.

Use the Client Credentials Grant to Obtain an Access Token

An access token is a credential that can be used by an application to access an API. To obtain an access token, the application accessing data remotely through the Web Service must connect to the StealthAUDIT token endpoint and use the Client ID and Client Secret to authenticate the access request. This is done using the Client Credentials grant. The Client Credentials grant is used when applications request an access token to access their own resources, not on behalf of a user. The following request parameters should be used:

- → `grant_type` (required) – The `grant_type` parameter must be set to `client_credentials`
- → `scope` (optional) – Your service may support different scopes for the client credentials grant

The client must then be authenticated for the request. Typically, the service will allow either additional request parameters, `client_ID` and `client_secret`, or accept those parameters in the HTTP Basic auth header.

The following example shows how to retrieve an access token.

```
POST /token HTTP/1.1

Host: authorization-server.com


grant_type=client_credentials

&client_id=xxxxxxxxxx

&client_secret=xxxxxxxxxx
```

RECOMMENDED: Tokens contain sensitive information and should be stored securely. See the Microsoft ConvertTo-SecureString article for additional information.

If the token does not have the ability to perform this request, is invalid, or the specific resource has been blocked from access remotely, an HTTP status code of 401 is returned.

If the request for an access token is valid, the authorization server generates an access token and returns it to the client. The following example shows a successful access token response.

```
HTTP/1.1 200 OK

Content-Type: application/json

Cache-Control: no-store

Pragma: no-cache


{

  "access_token":"MTQ0NjJkZmQ5OTM2NDE1ZTZjNGZmZjI3",

  "token_type":"bearer",

  "expires_in":3600,
```

```
  "refresh_token":"IwOGYzYTlmM2YxOTQ5MGE3YmNmMDFkNTVk",
  "scope":"create"
}
```

See the okta Access Token Response article for additional information on successful and unsuccessful responses to requests for access tokens.

The Client Secret expires after 72 hours. The access token expires after 7 days after which time you can request a refresh token.

Use the Client Credentials to Grant a Refresh Token

A refresh token contains the information required to obtain a renewed access token. Request a refresh token when the access token expires.

- → `grant_type` (required) – The `grant_type` parameter must be set to `client_credentials`
- → `refresh_token` (required) – The refresh token previously issue to the client
- → `scope` (optional) – The requested scope must not include additional scopes that were not issued in the original access token. If the scope is not included in the request, the service issues an access token with the same scope as previously issued.
- → Client Authentication – Required if the client was issued a secret

The authentication server then verifies the access request. If the request is valid, the service generates an access token.

The following example shows a refresh token grant.

```
POST /api/v1/token HTTP/1.1
Host: authorization-server.com

grant_type=refresh_token
&refresh_token=xxxxxxxxxxx
&client_id=xxxxxxxxxx
&client_secret=xxxxxxxxxx
```

If the token does not have the ability to perform this request, is invalid, or the specific resource has been blocked from access remotely, an HTTP status code of 401 is returned.

The response for a refresh token is the same as the response for an access token. Optionally, a new refresh token can be included in the response. If a new refresh token is not included in the response, the current refresh token will continue to be valid. The following example shows a successful access token response.

```
POST /oauth/token HTTP/1.1

Host: authorization-server.com


grant_type=refresh_token

&refresh_token=xxxxxxxxxxx

&client_id=xxxxxxxxxx

&client_secret=xxxxxxxxxx
```

See the okta Access Token Response article page for additional information on successful and unsuccessful responses to requests for access tokens.

The refresh token expires after 90 days.

Use the Access Token to Get Data from the StealthAUDIT Endpoint

Use the access token to call the API endpoints using PowerShell and retrieve data. The following tables provide additional information on retrieving data.

ROWS

This table provides information on how to call the REST API to retrieve data from a named table or view definition.

URL STRUCTURE	/api/v1/data/<object-name>/rows /api/v1/data/<alias-name>/rows
DESCRIPTION	Allows the caller to retrieve data from a table or view.
METHOD	GET, POST
PARAMETERS	**object-name** required value that specifies the unique object name. **alias-name** required value that specifies the unique alias associated with the table, available as a more thoughtfully designed namespace.

jobRuntimeKey *optional* the execution to retrieve information for – if this is omitted the latest report will be provided.

filters *optional* a filter to be applied prior to returning data, multiple filters are applied with "and" operators. If an array is specified for the value field for a filter, the filter will return any successful match from the array of values. String comparisons are case *insensitive*, below is a list of the available functions:

Filter functions
equals
not_equals
greater (*greater_equal*)
less (*less_equal*)
contains
starts_with

columns *optional* a list of columns to be returned, when not specified all columns will be returned. The columns specified by the groupby parameter should be omitted from this array.

groupby *optional* a list of columns to group each row by, resulting in a JSON object that contains those keys followed by an array of entries.

Sample JSON request

```
{
   jobRuntimeKey: "2018-11-05T13:15:30",
   columns: [ "url", "trusteeName", "rights" ],
   groupby: [ "hostName" ],
   filters: [
       {
           column: "hostName",
           function: "equals",
           value: "ENGINEERING01",
       },
```

	<pre>{ column: “trusteeName”, function: “equals”, value: [“Pete Smith”, “Jake Roberts”] }, …], }</pre>
RETURNS	A JSON array representation of the underlying table. **Sample JSON response** <pre>[{ hostName: “ENGINEERING01”, groupItems: [{ url: “https://site/list”, trusteeName: “Pete Smith”, rights: “Read” },] … }]</pre>
ERRORS	400 One or more the parameters passed in are invalid. 404 The object requested does not exist.

PROC

This table provides information on how to call the REST API to execute a stored procedure.

URL STRUCTURE	/api/v1/data/<object-name>/proc /api/v1/data/<alias-name>/proc
DESCRIPTION	Allows the caller to execute stored procedure and retrieve data.
METHOD	POST
PARAMETERS	**object-name** required value that specifies the unique object name. **groupby** *optional* a list of columns to group each row by, resulting in a JSON object that contains those keys followed by an array of entries. **Sample JSON request** `{` `  parameters: {` `  hostName: "SBNJENGINEERING01",` `  userName: "DOMAIN\pete.smith",` `  files: [` `  { name: "puppets.xls" },` `  { name: "groups.pdf" }` `  ]` `  },` `  groupby: [ "HostName" ],` `}` The parameters passed in here will be passed to the stored procedure untouched. Arrays are mapped to a user defined table type, currently only single value arrays are supported.
RETURNS	A JSON array representation of the underlying result data. **Sample JSON response** `[`

	<pre>{ hostName: "ENGINEERING01", groupItems: [{ url: "https://site/list", trusteeName: "Pete Smith", rights: "Read" },] … }]</pre>
ERRORS	400 One or more the parameters passed in are invalid. 404 The object requested does not exist.

PowerShell Commands for the REST API

The following examples show PowerShell commands commonly performed with the REST API.

Retrieve an Access Token

The following example shows how to retrieve an access token.

```
$body = @{
  client_id="[Insert Client ID Here]"
  client_secret="[Insert Client Secret Here]"
  grant_type="client_credentials"
}
```

```
$response = Invoke-WebRequest -Method POST -uri
   http://localhost:8082/api/v1/token -Body $body -ContentType
   "application/json"

$content = $response.Content | ConvertFrom-Json


$access_token = $content.access_token;

$refresh_token = $content.refresh_token;
```

RECOMMENDED: Tokens contain sensitive information and should be stored securely. See the Microsoft ConvertTo-SecureString article for additional information.

Retrieve Data from a Table or View

The following example shows how to retrieve data from a table or view.

```
$headers = @{

  Authorization="Bearer $access_token"

}

$response = Invoke-WebRequest -Method GET -uri
   http://localhost:8082/api/v1/data/SA_ADInventory_ComputersView/ro
   ws -Headers $headers

$content = $response.Content | ConvertFrom-Json
```

More Information

STEALTHbits is a data security software company. We help organizations ensure the right people have the right access to the right information. By giving our customers insight into who has access and ownership of their unstructured data, and protecting against malicious access, we reduce security risk, fulfill compliance requirements, and decrease operations expense.

For information on our products and solution lines, check out our website at www.stealthbits.com or send an email to our information center at info@stealthbits.com.

If you would like to speak with a STEALTHbits Sales Representative, please contact us at +1.201.447.9300 or via email at sales@stealthbits.com.

Have questions? Check out our online Documentation or our Training Videos (requires login): https://www.stealthbits.com/documentation. To speak to a STEALTHbits Representative: please contact STEALTHbits Support at +1.201.447.9359 or via email at support@stealthbits.com.

Need formal training on how to use a product more effectively in your organization? STEALTHbits is proud to offer FREE online training to all customers and prospects! For schedule information, visit: https://www.stealthbits.com/on-demand-training.