

The Lazy Man's Guide to the Tableau REST API

Created by out-of-work Keebler elves, not supported by Tableau or anyone else for that matter.

Using Postman	3
Installing Postman	3
Launching Postman.....	4
Postman UI	4
Postman Collections	4
API Reference	6
Sign In	6
Sign In as a Tableau Server Administrator.....	6
Sign In as a Tableau Server User.....	7
Payload Details	9
Error Conditions.....	9
Sign Out	10
Error Conditions.....	10
Query Datasources	11
Business Logic and Error Conditions.....	13
Query Datasource.....	13
Business Logic and Error Conditions.....	14
Error Conditions.....	15
Create Project.....	15
Payload Details	17
Error Conditions.....	17
Delete Project	17
Business Logic and Error Conditions.....	18
Update Project.....	18
Business Logic and Error Conditions.....	20
Add Tags to Workbook	20
Payload Details	22
Business Logic and Error Conditions.....	22
Delete Tag from Workbook	23
Business Logic and Error Conditions.....	24

Query Workbook	24
Query Workbook with a Preview Image.....	26
Payload Details	26
Business Logic and Error Conditions.....	27
Add Workbook to Favorites.....	28
Business Logic and Error Conditions.....	29
Delete Workbook from Favorites	30
Business Logic and Error Conditions.....	31
Query View with a Preview Image.....	31
Business Logic and Error Conditions.....	32
Query Views for Workbook	33
Query Views with Usage Statistics.....	34
Business Logic and Error Conditions.....	36
Query Workbook Preview Image.....	36
Business Logic and Error Conditions.....	37
Query Workbooks for User.....	38
Business Logic and Error Conditions.....	41
Add User to Site	41
Payload Details	43
Get Users on Site	43
Error Conditions.....	45
Remove User from Site.....	45
Business Logic and Error Conditions.....	46
Error Conditions.....	46
Create Site	47
Payload Details	48
Error Conditions.....	48
Update Site	49
Payload Details	50
Business Logic and Error Conditions.....	50
Query Site	51
Site Information Including Storage Usage	52
Error Conditions.....	54
Delete Site	54

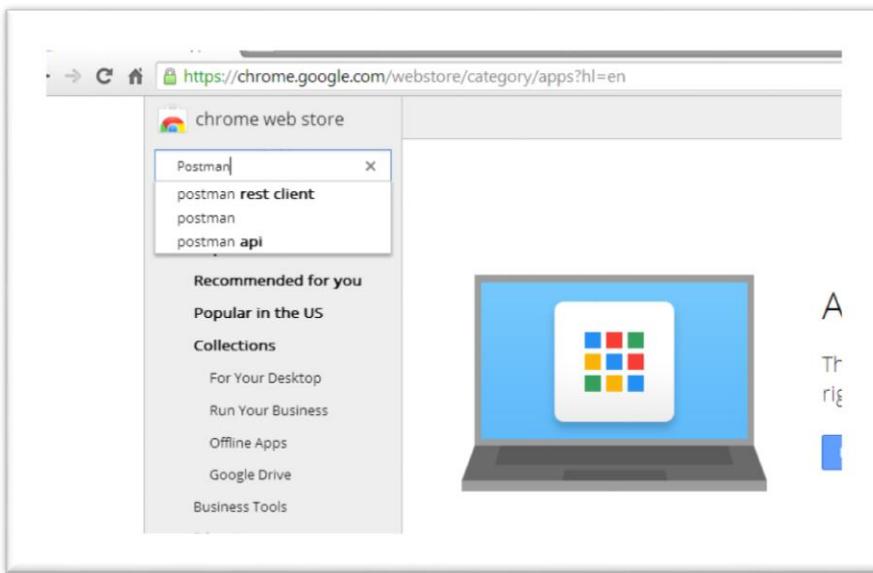
Using Postman

Postman is a free extension for the Google Chrome browser that you can use to call the Tableau REST API.

Using Postman, you can get a feel for how the API functions without having to write any code to make requests and handle responses. After you understand how a method works and have tested with Postman, you can write your production code with confidence.

Installing Postman

Download and install the Postman REST Client from the Google Chrome Web Store:



A screenshot of the Postman - REST Client page on the Google Chrome Web Store. The page has a header with the Postman logo, a rating of 4.5 stars (1945 reviews), and developer information. Below the header, there are tabs for 'OVERVIEW', 'DETAILS', 'REVIEWS', and 'RELATED'. The 'OVERVIEW' tab is selected, showing a screenshot of the Postman interface with various API endpoints listed. To the right of the screenshot, there is descriptive text about Postman's features, including 'Runs Offline'. A 'LAUNCH APP' button is located at the top right of the page.

You can also download it here:

<http://www.getpostman.com/>

Launching Postman

Postman does not install a button to chrome like many other extensions do. The easiest way to launch the tool is navigate to:

<chrome-extension://fdmmgilgnpjigdojopiooiodkmcomcm/index.html>

After the tool appears, bookmark it like you'd bookmark any other web page. You'll use the bookmark to re-launch the client in the future.

Postman UI

There are three important pieces of UI you need to be familiar with:



The **Headers** button shows and hides any headers that will included with a request. You will need to show Headers in order to add the **X-Tableau-Auth header** which is used for authentication against Tableau.

The **Method** drop-down controls whether you send POST, GET, PUT, or DELETE requests.

The **Raw Requests** window is where you add the REST API **Message Payload** for requests which require it.

A screenshot of the 'Raw' tab in Postman. The tab has four options: 'form-data', 'x-www-form-urlencoded', 'raw', and 'Text'. The 'raw' tab is selected and highlighted in grey. Below the tabs is a code editor containing XML code for a Tableau API request. The code is numbered from 1 to 5.

```
1 <tsRequest>
2   <credentials name="someUserName" password="somePassword">
3     <site contentUrl="" />
4   </credentials>
5 </tsRequest>
```

For additional assistance around making requests, please read [Postman help](#)

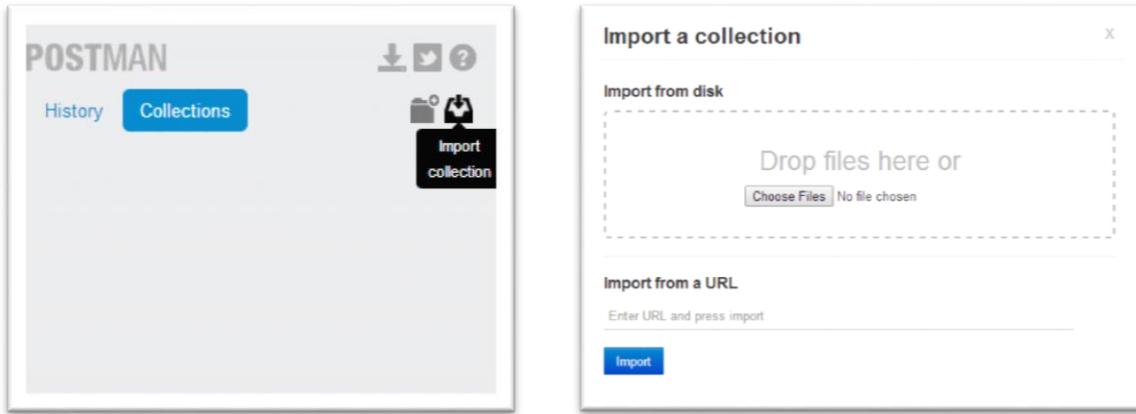
Postman Collections

Postman allows you to **save and re-use** requests after you define them. These requests can be organized into **Collections**. You can download a pre-created collection which includes sample requests for every API method here:

<http://1drv.ms/1kuf1Fp>

Note that this sample is not supported by Tableau in any shape or form. If you have problems with it or Postman you'll need to puzzle out things on your own.

To import the collection, click the **Import Collection button**, then drag the file JSON file you downloaded above into the **Drop Files Here** zone of Postman:



API Reference

With the Tableau Server REST API you can manage and change Tableau Server resources programmatically, via HTTP. The API gives you simple access to the functionality behind the data sources, projects, workbooks, site users, and sites on a Tableau server. You can use this access to create your own custom applications or to script interactions with Tableau Server resources.

Before you can use the Tableau Server REST API, you must enable it. See [Enabling the API](#) for steps.

Sign In

The screenshot shows a REST client interface with the following details:

- Authenticate: Sign-in**
- URI:** http://10.211.55.4/api/2.0/auth/signin
- Method:** POST (highlighted with a red box)
- Body:** XML payload (Text tab selected)

```
1 <tsRequest>
2   <credentials name="russch" password="REDACTED">
3     <site contentUrl="" />
4   </credentials>
5 </tsRequest>
```
- Response Headers:** Headers (15) - STATUS: 200 OK, TIME: 4030 ms
- Response Body:** XML response (Pretty tab selected)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse>
3   <xsi:linkType href="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api http://tableausoftware.com/api/ts-api-2_0_vsd.xsd">
4     <credentials token="6f19a9fd21aa99b577ffff6258e19a278">
5       <site contentUrl="" />
6     </credentials>
7   </tsResponse>
```

A red arrow points to the token value "6f19a9fd21aa99b577ffff6258e19a278" with the label "Authentication Token".

Sign In as a Tableau Server Administrator

Description	Signs you in as the Tableau Server system administrator. For production environments, it's recommended that you sign in via HTTPS.
Method and URI	POST /api/2.0/auth/signin
Message Payload	<pre><tsRequest> <credentials name="admin-username" password="admin-password" > <site contentUrl="target-site-content-URL" /> </credentials> </tsRequest></pre>

Response Code	200
Response Payload	<pre><tsResponse> <credentials token="admin-user-authenticity-token" > <site contentUrl="target-site-content-URL" /> </credentials> </tsResponse></pre>

Sign In as a Tableau Server User

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site in which the non-admin user lives (use [Query Site](#))
- Use the **Site ID** to list Users in the site (via [Get Users On Site](#))
- Pull the **User ID** that you need from the **Get Users On Site** response.
- Feed the User ID to the **id** attribute of the **user** element

The screenshot shows the Postman interface with the following details:

- URI:** `http://10.211.55.4/api/2.0/sites/russell?key=name`
- Method:** `GET`
- Auth Token:** `2cf275a018bc2ce39477b1d1cbcab45c`
- Headers:** X-Tableau-Auth
- Response Status:** 200 OK, Time: 221 ms
- Response Body (XML):**

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse>
3   <site id="c174d512-a1fd-4cc0-b750-2f67b1cb877c" name="russell" contentUrl="russellSite" adminMode="ContentAndUsers" userQuota="11"
4     <xsi:schemaLocation="http://tableausoftware.com/api/xsd" xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance" />
5   </site>
6 </tsResponse>
    
```

A red arrow points to the `id="c174d512-a1fd-4cc0-b750-2f67b1cb877c"` attribute in the XML response, labeled "Site ID".

User Management: Get Users On Site

Get all users in site

Site ID

Method

GET

X-Tableau-Auth Auth Token 2cf275a018bc2ce39477b1d1cbcab45c

Header Value

Send Save Preview Add to collection Reset

Body Headers (15) STATUS 200 OK TIME 260 ms

Pretty Raw Preview JSON XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
  http://tableausoftware.com/api/ts-api-2.0.xsd">
5   <users>
6     <user id="e04d08d7-4455-4975-9662-3fe1a013df74" name="Paul" role="Interactor" publish="false" contentAdmin="false" lastLogin=""
7       externalAuthUserId="" />
8     <user id="feba71c8-5b08-4a17-9819-5c3081d2e7f7" name="russch" role="Interactor" publish="false" contentAdmin="false" lastLogin="2014-05-
9       31T22:18:46Z" externalAuthUserId="" />
10   </users>
11 </tsResponse>
12

```

User ID

Authenticating: Sign in (as Tableau User)

Impersonate a Tableau User

http://10.211.55.4/api/2.0/auth/signin

Site contentURL (Called "Site ID" in Tableau' portal)

POST

Header Value

Manage presets

form-data x-www-form-urlencoded raw Text

```

1 <tsRequest>
2   <credentials name="russch" password="*****" />
3   <site contentUrl="russellsite" />
4   <user id="e04d08d7-4455-4975-9662-3fe1a013df74" />
5 </credentials>
6 </tsRequest>

```

Paul's User Id

Send Save Preview Add to collection Reset

Body Headers (15) STATUS 200 OK TIME 156 ms

Pretty Raw Preview JSON XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
  http://tableausoftware.com/api/ts-api-2.0.xsd">
5   <credentials token="d15b5360cacf2ddb261354688803082e">
6     <site contentUrl="russellsite" />
7   </credentials>
8 </tsResponse>
9

```

Access token to login as Paul

Description	Signs you in as a Tableau Server user. For production environments, it's recommended that you sign in via HTTPS.
-------------	--

Method and URI	POST /api/2.0/auth/signin
Message Payload	<pre><tsRequest> <credentials name="admin-username" password="admin-password" > <site contentUrl="target-site-content-URL" /> <user id="user-to-run-as" /> </credentials> </tsRequest></pre>
Response Code	200
Response Payload	<pre><tsResponse> <credentials token="run-as-user-token" > <site contentUrl="target-site-content-URL" /> </credentials> </tsResponse></pre>

Payload Details

The `<site>` element is required in the request to specify the site that the user will be logged into. When the `<site>` element is empty in the request, the default site will be used:

```
<site />
```

The default site can also be specified using empty quotes:

```
<site contentUrl="" />
```

The `<site>` element is always populated in the response.

Error Conditions

HTTP Response Code	Condition	Details
401	Login error	The name or password is invalid for the given site, or the site content URL is invalid.
403	Non-admin sign-in forbidden	An attempt was made to sign in a user (non-administrator) directly, using the user's credentials. A Tableau Server user can be signed in only indirectly via an administrator's credentials.
404	User not found	The user ID given in the payload for a user sign-in does not correspond to an existing user on the site.

405	Invalid request method	Request type was something other than a POST.
-----	------------------------	---

Sign Out

What you need:

- The **X-Tableau-Auth** token that was issued to the user you want to logout

The screenshot shows a 'Sign Out' request being configured in a tool like Postman or curl. The 'URI' field contains `http://10.211.55.4/api/2.0/auth/signout`. The 'Auth Token' field contains the value `965b74780470df536651f4207e24882b`. The 'Method' dropdown is set to 'POST'. The 'Body' section is set to 'Text' and contains the number '1'. The response status is shown as '204 No Content' with a time of '107 ms'. The response payload is 'Status code: Successful logout!'. Red boxes highlight the 'URI', 'Auth Token', 'Method', and the response message in the body.

Description	Signs you out of the current session and invalidates the token.
Method and URI	POST /api/2.0/auth/signout
Message Payload	None
Response Code	204
Response Payload	None

Error Conditions

HTTP Response Code	Condition	Details
--------------------	-----------	---------

405	Invalid request method	Request type was something other than a POST.
-----	------------------------	---

Query Datasources

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site in which your data source lives (via [Query Site](#))
- Use the **Site ID** to call **Query Datasources**

The screenshot shows a cURL-like interface for making API requests. The URL field contains `http://10.211.55.4/api/2.0/sites/94f96a34-64e7-49f2-8624-e77426bbd490/datasources`. The Method dropdown is set to `GET`. The X-Tableau-Auth header is set to `Auth Token 8987dee32b1e0e770c663f481c87680a`. The response body is displayed below, showing XML output for multiple data sources, with several IDs highlighted in red boxes.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
      http://tableausoftware.com/api/ts-api-2.0.xsd">
5   <pagination pageNumber="1" pageSize="100" totalAvailable="4"/>
6   <datasources>
7     <datasource id="efa4144c-89ec-11e3-9d48-4f3cc1acc9a7" name="SSAS Cube - AdventureWorks" type="msolap">
8       <project id="er4rc44-89ec-11e3-972e-0be00158ccb0" name="Default"/>
9       <tags></tags>
10    </datasource>
11    <datasource id="efa4144c-89ec-11e3-87fb-13b11e71a593" name="ERP - Cost of Goods" type="csv">
12       <project id="er4rc44-89ec-11e3-972e-0be00158ccb0" name="Default"/>
13       <tags></tags>
14    </datasource>
15    <datasource id="efa4144c-89ec-11e3-afc1-d7c0b466da94" name="I'm a test" type="excel">
16       <project id="er4rc44-89ec-11e3-972e-0be00158ccb0" name="Default"/>
17       <tags></tags>
18    </datasource>
19    <datasource id="32573d84-e1eb-4b61-9d9b-5574bc33349f" name="Junction Table Inline w DS Filter" type="sqlserver">
20       <project id="er4rc44-89ec-11e3-972e-0be00158ccb0" name="Default"/>
21       <tags></tags>
22    </datasource>
23  </datasources>
24 </tsResponse>

```

Description	Returns a list of data sources on the specified site, with optional parameters for specifying the paging of large results.
Method and URI	GET /api/2.0/sites/ site-id /datasources

	GET /api/2.0/sites/ site-id /datasources?pageSize= page-size &pageNumber= page-number
Message Payload	None
Response Code	200
Response Payload	<pre> <tsResponse> <pagination pageNumber="pageNumber" pageSize="page-size" totalAvailable="total-available" /> <datasources> <datasource id="datasource1-id" name="datasource1-name" type="datasource1-type" > <project id="project-id" name="project-name" /> <tags> <tag label="tag1"/> <tag label="tag2"/> ... more tags ... </tags> </datasource> <datasource id="datasource2-id" name="datasource2-name" type="datasource2-type" > <project id="project-id" name="project-name" /> <tags> <tag label="tag3"/> ... more tags ... </tags> </datasource> ... more datasources ... </datasources> </tsResponse></pre>

Business Logic and Error Conditions

Non-Administrator User Access

Tableau Server users who are not administrators can access this API method, but they can only view data sources for which they have the **Connect** capability allowed (either explicitly or implicitly).

Pagination Parameter Defaults

By default the page size is 100 and page 1 is served. One or both parameters can be overridden explicitly.

Error Conditions

HTTP Response Code	Condition	Details
400	Invalid page size	The page size parameter is not an integer, or is less than one.
400	Invalid page number	The page number parameter is not an integer, is less than one, or is greater than the final page number for data sources at the requested page size.
404	Site not found	The site ID given in the URI does not correspond to an existing site.
405	Invalid request method	Request type was something other than a GET.

Query Datasource

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site in which your data source lives ([Query Site](#))
- Use the **Site ID** to call [**Query Datasources**](#)
- Grab a specific **Datasource ID** and use it with [**Query Datasource**](#)

Datasources: Query Datasource

Query a specific data source

Uri, including Site ID and Datasource ID	Method
http://10.211.55.4/api/2.0/sites/94f96a34-64e7-49f2-8624-e77426bbd490/datasources/efa4144c-89ec-11e3-9d48-4f3c0	GET
X-Tableau-Auth	Auth Token
8987dee32b1e0e770c663f481c87680a	<input type="button" value="X"/>
Manage presets	
Header Value	
Send	Save
Preview	Add to collection
Reset	

Body Headers (15) STATUS 200 OK TIME 170 ms

Pretty Raw Preview JSON XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
5     http://tableausoftware.com/api/ts-api-2.0.xsd">
6   <datasource id="efa4144c-89ec-11e3-9d48-4f3cc1acc9a7" name="SSAS Cube - AdventureWorks" type="msolap">
7     <project id="efaffc44-89ec-972e-0bebfb58ccb0" name="Default"/>
8   </datasource>
9 </tsResponse>
10

```

Description	Returns metadata about a specific data source.
Method and URI	GET /api/2.0/sites/site-id/datasources/ datasource-id
Message Payload	None
Response Code	200
Response Payload	<tsResponse> <datasource id="datasource-id" name="some-name" type="some-type" > <project id="project-id" name="project-name" /> <tags> <tag label="tag1"/> <tag label="tag2"/> ... more tags ... </tags> </datasource> </tsResponse>

Business Logic and Error Conditions

Non-Administrator User Access

Tableau Server users who are not administrators can access this API method, but they can only view data sources for which they have the **Connect** capability allowed (either explicitly or implicitly).

Error Conditions

Condition	HTTP Response Code	Details
403	Read forbidden	A non-administrator user invoked the API method and attempted to query a data source for which he or she did not have the Connect capability allowed.
404	Site not found	The site ID given in the URI does not correspond to an existing site.
404	Data source not found	The data source ID given in the URI does not correspond to an existing data source.
405	Invalid request method	Request type was something other than a <code>GET</code> .

Create Project

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) in which you want to create the Project
- Use the **Site ID** to call **Create Project**

Projects. Create Project

Create a new project in a specific site

Uri, including Site Id

http://10.211.55.4/api/2.0/sites/94f96a34-64e7-49f2-8624-e77426bbd490/projects

Method

POST

X-Tableau-Auth Auth Token

8987dee32b1e0e770c663f481c87680a

Header

Value

form-data **x-www-form-urlencoded** **raw** **Text**

```

1 <tsRequest>
2   <project name="Hello World"
3     description="A new project created with the REST API" />
4 </tsRequest>
5

```

Send **Save** **Preview** **Add to collection** **Reset**

Body Headers (14) STATUS 201 Created TIME 200 ms

Pretty Raw Preview  JSON XML

Info about new Project

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse>
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
      http://tableausoftware.com/api/ts-api-2.0.xsd">
5     <project id="f22ffc0e-74da-4c13-924f-f304b65d949b" name="Hello World" description="A new project created with the REST API"/>
6   </tsResponse>
7

```

Description	Creates a project on the specified site.
Method and URI	POST /api/2.0/sites/ site-id /projects
Message Payload	<pre> <tsRequest> <project name="project-name" description="project-description" /> </tsRequest></pre>
Response Code	201
Response Payload	<pre> <tsResponse> <project id="new-project-id" name="project-name" description="project-description" /> </tsResponse></pre>
Response Headers	Location: /api/2.0/sites/site-id/projects/new-project-id

Payload Details

The `project-name` attribute is required for this operation; the `project-description` attribute is optional.

Error Conditions

HTTP Response Code	Condition	Details
404	Site not found	The site ID given in the URI does not correspond to an existing site.
405	Invalid request method	Request type was something other than a <code>POST</code> .
409	Project name conflict	The project name in the request already belongs to the given site in the system. For the purpose of uniqueness checks, project names are case-insensitive.

Delete Project

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) from which you want to delete the Project
- Get the **Project ID** of the Project you wish to delete (There is no REST API method which allows you do this directly. Use other method like [Query Datasources](#) or [Query Workbook](#) to get at the information indirectly. If you have access to the workgroup database, you can also pull this information from the **projects** table (`projects.luid`)
- Use the **Project ID** to call **Delete Project**

Projects: Delete Project

Delete a project Uri, including Site ID and Project ID Method

http://10.211.55.4/api/2.0/sites/94f96a34-64e7-49f2-8624-e77426bbd490/projects/f22ffc0e-74da-4c13-924f-f3d4b65d9 **DELETE**

X-Tableau-Auth **Auth Token** 8987dee32b1e0e770c663f481c87680a

Header Value

form-data x-www-form-urlencoded raw **Text**

1

Send **Save** **Preview** **Add to collection** **Reset**

Body Headers (12) **STATUS** 204 No Content **TIME** 314 ms

Pretty Raw Preview **JSON** **XML**

1 Status: Success!

Business Logic and Error Conditions

When a project is deleted in the system all of its assets are also deleted: associated workbooks, datasources, project view options, and rights. Use this method with caution.

Error Conditions

HTTP Response Code	Condition	Details
403	Deletion forbidden	Attempt to delete the default project, which cannot be deleted.
404	Site not found	The site ID or URL namespace given in the URI does not correspond to an existing site.
404	Project not found	Either the project ID given in the URI does not correspond to an existing project or the project is not found on this site.
405	Invalid request method	Request type was something other than a DELETE .

Update Project

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) in which you want to update the Project

- Get the **Project ID** of the Project you wish to delete (There is no REST API method which allows you do this directly. Use other method like [Query Datasources](#) or [Query Workbook](#) to get at the information indirectly. If you have access to the workgroup database, you can also pull this information from the **projects** table (projects.luid)
- Use the **Project ID** to call **Update Project**

Projects: Create Project

Create a new project in a specific site

Uri with Site ID and Project ID: http://10.211.55.4/api/2.0/sites/94f96a34-64e7-49f2-8624-e77426bbd490/projects/f22ffc0e-74da-4c13-924f-f3d4b65d9
Method: PUT

X-Tableau-Auth: Auth Token: 8987dee32b1e0e770c663f481c87680a

Header: Value

Text (selected):

```

1 <tsRequest>
2   <project description="A new project created with the REST API which has been UPDATED" />
3 </tsRequest>
4

```

Send, **Save**, **Preview**, **Add to collection**, **Reset**

Body: Headers (15), STATUS 200 OK, TIME 129 ms

Pretty, Raw, Preview, JSON, XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse>
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
  http://tableausoftware.com/api/ts-api-2.0.xsd">
5     <project id="f22ffc0e-74da-4c13-924f-f3d4b65d949b" name="Hello World" description="A new project created with the REST API which has been
  UPDATED"/>
6   </tsResponse>

```

Modified Description

Description	Updates the name or description of the specified project.
Method and URI	PUT /api/2.0/sites/site-id/projects/ project-id
Message Payload	<tsRequest> <project name="some-new-name" description="some-new-description" /> </tsRequest>
Response Code	200
Response Payload	<tsResponse> <project name="some-name"

```

    description="some-description" />
</tsResponse>

```

Business Logic and Error Conditions

Description	Deletes the specified project on a specific site.
Method and URI	DELETE/api/2.0/sites/ site-id /projects/ project-id
Message Payload	None
Response Code	204
Response Payload	None

Payload Variations

Any combination of the attributes inside the `<project>` element is valid. Only those attributes present will result in updates to their values in the project. If no attributes are present, then the update won't have an effect.

HTTP Response Code	Condition	Details
403	Update forbidden	Attempt to rename the default project.
404	Site not found	The site ID given in the URI does not correspond to an existing site.
404	Project not found	The project ID given in the URI does not correspond to an existing project.
404	Project ID mismatch	The payload contains a project ID (which is optional) and it does not match up with the ID in the URI.
405	Invalid request method	Request type was something other than a <code>PUT</code> .
409	Project name conflict	The project name in the request already belongs to the given site in the system. For the purpose of uniqueness checks, project names are case-insensitive.

Add Tags to Workbook

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) in which your workbook lives
- Use the **Site ID** to list Users in the site (via [Get Users On Site](#))

- Pick a **User ID** and feed it to [Query Workbooks For User](#) in order to get a list of workbooks owned by this user
- Grab the **Workbook ID** of your choice and feed it to [Add Tags to Workbook](#)

Workbook Management: Add Tags to a Workbook

Add one or more tags to a workbook [Uri, includes Site ID & Workbook ID](#) Method

Method: PUT Headers (1)

X-Tableau-Auth Auth Token: 8987dee32b1e0e770c663f481c87680a

Header Value

Manage presets

form-data x-www-form-urlencoded raw Text

```

2   <workbook id="efbbe45a-89ec-11e3-8e3a-1bae9933f8ef">
3     <tags>
4       <tag label="Tagged by the REST API" />
5       <tag label="tag2" />
6     </tags>
7   </workbook>
8 </tsRequest>
9

```

Include Workbook ID here, too

Send Save Preview Add to collection Reset

Body Headers (15) STATUS 200 OK TIME 349 ms

Pretty Raw Preview

JSON XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tResponse version="2.0"
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
5   http://tableausoftware.com/api/ts-api-2.0.xsd">
6     <workbook id="efbbe45a-89ec-11e3-8e3a-1bae9933f8ef" name="Real Estate">
7       <project id="efaffc44-89ec-11e3-8cc9-c7a81e1f7f4d" name="Tableau Samples"/>
8       <tags>
9         <tag label="API"/>
10        <tag label="tag2"/>
11        <tag label="Tagged"/>
12        <tag label="by"/>
13        <tag label="the"/>
14        <tag label="REST"/>
15     </tags>
16     <views>
17       <view id="f01d86a6-89ec-11e3-a04c-7f2aac50fa01"/>
18       <view id="f01d86a6-89ec-11e3-8c16-9bae1570d152"/>
19       <view id="f01d86a6-89ec-11e3-a876-87188651a4fd"/>
20       <view id="f01d86a6-89ec-11e3-9f4c-17a6772f6520"/>
21       <view id="f01d86a6-89ec-11e3-82ad-2f5c1bb5974e"/>
22     </views>
23   </workbook>

```

Note how multiple words in a single "Tag String" (above) are broken down into single tags

Description	Adds one or more tags to the specified workbook.
Method and URI	PUT /api/2.0/sites/site-id/workbooks/ workbook-id
Message Payload	<pre> <tsRequest> <workbook id="workbook-id"> <tags> <tag label="tag1" /> <tag label="tag2" /> ... more tags ... </tags> </workbook> </pre>

	</tsRequest>
Response Code	200
Response Payload	<pre> <tsResponse> <workbook id="workbook-id" name="workbook-name" > <project id="project-id" name="project-name" /> <tags> <tag label="tag1"/> <tag label="tag2"/> ... more tags ... </tags> <views> <view id="view1-id" /> <view id="view2-id" /> ... more views ... </views> </workbook> </tsResponse> </pre>

Payload Details

If the workbook is already tagged with any tags in the payload, then those tags are ignored and the workbook retains them. If the tags element is empty, no new tags are added to the workbook.

Business Logic and Error Conditions

Non-Administrator Access

Tableau Server users who are not administrators can use this method but they can only add a tag to workbook for which they have the **Add Tag** capability allowed (either explicitly or implicitly).

HTTP Response Code	Condition	Details

403	Adding tags forbidden	A Tableau Server user who isn't an administrator invoked the API method and attempted to add a tag to a workbook for which he or she did not have the Add Tag capability allowed. This error is raised even in the case when the <code>tags</code> element is empty.
404	Site not found	The site ID given in the URI is not for an existing site.
404	Workbook not found	The workbook ID given in URI does not correspond to an existing workbook.
404	Workbook ID mismatch	The payload contains a workbook ID (which is optional) and it does not match the ID in the URI.
405	Invalid request method	Request type was something other than a <code>PUT</code> .

Delete Tag from Workbook

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) in which your workbook lives
- Use the **Site ID** to list Users in the site (via [Get Users On Site](#))
- Pick a **User ID** and feed it to [Query Workbooks For User](#) in order to get a list of workbooks owned by this user
- Grab the **Workbook ID** of your choice and feed it to **Delete Tag from Workbook**

Workbook Management: Delete Tag from a Workbook

Delete a tag from the specified workbook **Uri, including Site ID, Workbook ID, and Tag to be deleted**

/api/2.0/sites/94f96a34-64e7-49f2-8624-e77426bb490/workbooks/efbbe45a-89ec-11e3-8e3a-1bae9933f8ef/tags/tagd

DELETE

Method

X-Tableau-Auth Auth Token: 8987dee32b1e0e770c663f481c87680a

Header Value

form-data x-www-form-urlencoded raw Text

2
3
4
5
6
7
8
9

Send Save Preview Add to collection Reset

Body Headers (12) STATUS 204 No Content TIME 209 ms

Pretty Raw Preview JSON XML

1 Success!

Description	Deletes one or more tags from the specified workbook.
-------------	---

Method and URI	DELETE /api/2.0/sites/site-id/workbooks/ workbook-id /tags/ tag-name
Message Payload	None
Response Code	204
Response Payload	None

Business Logic and Error Conditions

Non-Admin User Access

Tableau Server users who are not administrators can access this API method, but they can only delete a tag from a workbook for which they have the capability implicitly allowed (in other words, as owner of the workbook).

HTTP Response Code	Condition	Details
403	Deleting tags forbidden	A Tableau Server user who isn't an administrator invoked the API method and attempted to delete a tag from a workbook for which he or she did not have the capability implicitly allowed.
404	Site not found	The site ID or URL namespace given in the URI does not correspond to an existing site.
404	Workbook not found	The workbook ID given in the URI does not correspond to an existing workbook.
404	Tag not found	The tag given in the URI does not exist for the given workbook.
405	Invalid request method	Request type was something other than a <code>DELETE</code> .

Query Workbook

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) in which the workbook lives
- Use the **Site ID** to list Users in the site (via [Get Users On Site](#))
- Pick a **User ID** and feed it to [Query Workbooks For User](#) in order to get a list of workbooks owned by this user
- Grab the **Workbook ID** of your choice and feed it to [Query Workbook](#)

WORKBOOK MANAGEMENT: Query WORKBOOK

Get the details of a specific workbook

Uri, including Site ID & Workbook ID

Method

3.211.55.4/api/2.0/sites/94f96a34-64e7-49f2-8624-e77426bbd490/workbooks/efbbe45a-89ec-11e3-8e3a-1bae9933f8ef

GET

Headers (1)

X-Tableau-Auth Auth Token: 7f664407d46b0f512cc46af964c2b40b

Manage presets

Header Value

Send Save Preview Add to collection Reset

Body Headers (15) STATUS 200 OK TIME 127 ms

Pretty Raw Preview JSON XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse version="2.0"
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
      http://tableausoftware.com/api/ts-api-2.0.xsd">
5     <workbook id="efbbe45a-89ec-11e3-8e3a-1bae9933f8ef" name="Real Estate">
6       <project id="efaffc44-89ec-11e3-8cc9-c7a81e1f7f4d" name="Tableau Samples"/>
7       <tags>
8         <tag label="API"/>
9         <tag label="Tagged"/>
10        <tag label="by"/>
11        <tag label="REST"/>
12        <tag label="the"/>
13      </tags>
14      <views>
15        <view id="f01d86a6-89ec-11e3-a04c-7f2aac50fa01"/>
16        <view id="f01d86a6-89ec-11e3-8c16-9bae157bd152"/>
17        <view id="f01d86a6-89ec-11e3-a876-87188651a4fd"/>
18        <view id="f01d86a6-89ec-11e3-9f4c-17a6772f6520"/>
19        <view id="f01d86a6-89ec-11e3-82ad-2f5c1bb5974e"/>
20      </views>
21    </workbook>
22  </tsResponse>
23

```

Description	Returns metadata about the specified workbook, optionally including a thumbnail.
Method and URI	<p>GET /api/2.0/sites/site-id/workbooks/workbook-id</p> <p>GET /api/2.0/sites/site-id/workbooks/workbook-id?previewImage=false</p>
Message Payload	None
Response Code	200
Response Payload	<tsResponse> <workbook id=" workbook-id "> name="some-name" description="some-description" > <project id="project-id"> name="project-name" /> <tags> <tag label="tag1"/>

```

<tag label="tag2"/>

... more tags ...

</tags>

<views>

<view id="view1-id"/>

<view id="view2-id"/>

... more views ...

</views>

</workbook>

</tsResponse>

```

Query Workbook with a Preview Image

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) in which the workbook lives
- Use the **Site ID** to list Users in the site (via [Get Users On Site](#))
- Pick a **User ID** and feed it to [Query Workbooks For User](#) in order to get a list of workbooks owned by this user
- Grab the **Workbook ID** of your choice and feed it to [Query Workbook with a Preview Image](#)

Description	Returns metadata about the specified workbook, including a thumbnail.
Method and URI	GET /api/2.0/sites/site-id/workbooks/ workbook-id ?previewImage=true
Message Payload	None
Response Code	200
Response Payload	Same as above embedded in a MIME multipart message with the following format: MIME header XML payload (in the format shown in the above table) The binary preview image.

Payload Details

An example of the full response format follows. The MIME multipart message is shown split by parts via a table, and the "interpart boundary" text is omitted as the table rows act to show where this delimiter would occur.

```
MIME-Version: 1.0  
Content-Type: multipart/mixed; boundary=interpartBoundary  
  
Content-Type: text/xml  
  
<tsResponse>  
  <workbook>  
    <!-- see markup example from previous section -->  
  </workbook>  
</tsResponse>
```

```
Content-Disposition: name="myPreviewImage"; filename="foo"  
Content-Transfer-Encoding: binary  
Content-Type: application/octet-stream  
...the-tableau-workbook-preview-image-goes-here...
```

Business Logic and Error Conditions

Non-Administrator User Access

Tableau Server users can access this API method, but they can only query workbooks for which they have the **Read** capability allowed (either explicitly or implicitly).

HTTP Response Code	Condition	Details
400	Invalid URI parameters	The <code>previewImage</code> parameter is specified but it has a value other than true or false.
403	Read forbidden	A user who isn't an administrator invoked the API method and attempted to query workbooks for which he or she does not have the Read capability allowed.
404	Workbook not found	The workbook ID given in the URI does not correspond to an existing workbook.
404	Site not found	The site ID given in the URI does not correspond to an existing site.
405	Invalid request method	Request type was something other than a <code>GET</code> .

Add Workbook to Favorites

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) in which your workbook lives
- Use the **Site ID** to list Users in the site (via [Get Users On Site](#))
- Pick a **User ID** and feed it to [Query Workbooks For User](#) in order to get a list of workbooks owned by this user
- Grab the **Workbook ID** of your choice and feed it (along with **Site ID** and **User ID**) to **Add Workbook to Favorites**

The screenshot shows the Postman interface for making a REST API call. The URL is set to `http://10.211.55.4/api/2.0/sites/94f96a34-64e7-49f2-8624-e77426bbd490/favorites/1d0f2b7e-6594-4105-a179-128b30d`. The method is set to `PUT`. The `Auth Token` header is set to `be96cb5e71e471397b38443aea16a45i`. The body contains the XML payload for favoriting a workbook:

```
1 <tsRequest>
2   <favorite label="I was Favorited by REST">
3     <workbook id="efbbe45a-89ec-11e3-8e3a-1bae9933f8ef" />
4   </favorite>
5 </tsRequest>
```

The `Workbook ID` is highlighted in red. Below the body, the response status is `200 OK` with a time of `175 ms`. The response body is shown in XML format:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse version="2.0"
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
      http://tableausoftware.com/api/ts-api-2.0.xsd">
5   <favorites>
6     <favorite label="I was Favorited by REST">
7       <workbook id="efbbe45a-89ec-11e3-8e3a-1bae9933f8ef"/>
8     </favorite>
9   </favorites>
10 </tsResponse>
```

Description	Adds the specified workbook to a user's favorites,
Method and URI	PUT /api/2.0/sites/site-id/favorites/ user-id
Message Payload	<tsRequest>

	<pre> <favorite label="favorite-label"> <workbook id="workbook-id" /> </favorite> </tsRequest> </pre>
Response Code	200
Response Payload	<pre> <tsResponse> <favorites> <favorite label="favorite-label"> <workbook id="workbook-id" /> </favorite> <favorite label="favorite2"> <view id="view-id" /> </favorite> more favorites ... </favorites> </tsResponse> </pre>

If the user already has the workbook in his or her favorites with the same label, then the operation will have no effect. If the label differs, then the original favorite is overwritten.

Business Logic and Error Conditions

Non-Administrator User Access

Tableau Server users can access this API method, but they can only add a workbook to a Favorites list if they have the **Add Favorite** capability allowed for the workbook (either explicitly or implicitly).

HTTP Response Code	Condition	Details
400	Invalid label	The favorite label is empty or consists of only white space characters.
403	Access to Favorites forbidden	A Tableau Server user invoked the API method and attempted to add a workbook to their Favorites list without the Add Favorite capability. This will always be the case when the user ID in the URI identifies a user other than the user invoking the method.
404	User not found	The user ID given in the URI does not correspond to an existing user.

404	Site not found	The site ID given in the URI does not correspond to an existing site.
404	Workbook not found	The workbook ID given in the payload does not correspond to an existing workbook.
405	Invalid request method	Request type was something other than a PUT.
409	Label conflict	There is already a favorite with the same label for a different workbook in the given user's favorites.

Delete Workbook from Favorites

What you need:

- Sign In to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) in which your workbook lives
- Use the **Site ID** to list Users in the site (via [Get Users On Site](#))
- Pick a **User ID** and feed it to [Query Workbooks For User](#) in order to get a list of workbooks owned by this user
- Grab the **Workbook ID** that is favorited and feed it (along with **Site ID** and **User ID**) to [Delete Workbook to Favorites](#)

The screenshot shows a POSTMAN interface with the following details:

- Method:** DELETE
- URL:** `77426bbd490/favorites/fd0f2b7e-6594-4105-a179-128b30d42c79/workbooks/efbbe45a-89ec-11e3-8e3a-1bae9933f8ef`
- Headers:**
 - X-Tableau-Auth: **Auth Token** (highlighted with a red box)
 - Site ID: **User ID** (highlighted with a red box)
- Body:** (empty)
- Response:**
 - Status: 204 No Content
 - Time: 87 ms
 - Message: "Favorite Deleted!" (highlighted with a red box)

Description	Deletes a workbook from a user's favorites.
Method and URI	DELETE /api/2.0/sites/site-id/favorites/user-id/workbooks/workbook-id
Message Payload	None

Response Code	204
Response Payload	None

Business Logic and Error Conditions

If the workbook you specify isn't a Favorite, then this call has no effect.

Non-Administrator User Access

Tableau Server users can access this API method, but they can only delete a workbook from their own Favorites list.

Error Conditions

HTTP Response Code	Condition	Details
403	Access to Favorites forbidden	A Tableau Server user invoked the API method and attempted to delete a workbook from a different user's favorites..
404	User not found	The user ID given in the URI does not correspond to an existing user.
404	Site not found	The site ID given in the URI does not correspond to an existing site.
404	Workbook not found	The workbook ID given in the payload does not correspond to an existing workbook.
404	Workbook not a favorite	The workbook ID given in the URI exists but is not a favorite of the given user.
405	Invalid request method	Request type was something other than a DELETE.

Query View with a Preview Image

What you need:

- **Sign In** to get a X-Tableau-Auth token
- Get the **Site ID** of the site (via [Query Site](#)) in which your view lives
- Use the **Site ID** to list Users in the site (via [Get Users On Site](#))
- Pick a **User ID** and feed it to [Query Workbooks For User](#) in order to get a list of workbooks owned by this user
- Grab the **Workbook ID** and pass it to [Query Views for Workbook](#)
- Use the **View ID** of your choice (and along with **Site ID** and **Workbook ID**) pass to [Query View with a Preview Image](#)

The screenshot shows a POSTMAN interface with a request to `10.211.55.4/api/auth?apiVersion=2.0&destinationId=77426bbd480/workbooks/efbbe45a-89ec-11e3-8e3a-1bae9933f8ef/views/f01d86a6-89ec-11e3-a04c-712aac50fa01/previewImage`. The request includes `X-Tableau-Auth` and `Auth Token` headers. The response status is `200 OK`. A screenshot of Charles proxy shows the raw response structure and a thumbnail image.

Description	Return the thumbnail for a specific view.
Method and URI	GET <code>/api/2.0/sites/site-id/workbooks/workbook-id/views/view-id/previewImage</code>
Message Payload	None
Response Code	200
Response Payload	The view's preview image in PNG format (MIME media type <code>image/png</code>).

Business Logic and Error Conditions

Non-Administrator User Access

Tableau Server users can access this API method, but they can only query workbook views for which they have the **Read** capability allowed (either explicitly or implicitly).

Error Conditions

HTTP Response Code	Condition	Details

403	Read forbidden	A Tableau Server user invoked the API method and attempted to query a workbook view for which he or she did not have the Read capability allowed.
404	View not found	The view ID given in the URI does not correspond to an existing view in the given workbook.
404	Site not found	The site ID given in the URI does not correspond to an existing site.
404	Workbook not found	The workbook ID given in the payload does not correspond to an existing workbook.
405	Invalid request method	Request type was something other than a _{DELETE} .

Query Views for Workbook

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) in which your view lives
- Use the **Site ID** to list Users in the site (via [Get Users On Site](#))
- Pick a **User ID** and feed it to [Query Workbooks For User](#) in order to get a list of workbooks owned by this user
- Grab the **Workbook ID** and pass [Query Views for a Workbook](#)

WORKBOOK MANAGEMENT: Query Views for a Workbook

Get a list of views in a Workbook Uri, includes Site ID, Workbook ID

Method: GET

5.4/api/2.0/sites/94f96a34-64e7-49f2-8624-e77426bbd490/workbooks/efbbe45a-89ec-11e3-8e3a-1bae9933f8ef/views/

X-Tableau-Auth Auth Token: be96cb5e71e471397b38443aea16a45i

Header: Value

Send Save Preview Add to collection Reset

Body Headers (14) STATUS 200 OK TIME 87 ms

Pretty Raw Preview JSON XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
  http://tableausoftware.com/api/ts-api-2.0.xsd">
5     <views>
6       <view id="f01d86a6-89ec-11e3-9f4c-17a6772f6520" name="Real Estate Market Report"/>
7       <view id="f01d86a6-89ec-11e3-a876-87188651a4fd" name="Seattle Area Sales Activity"/>
8       <view id="f01d86a6-89ec-11e3-82ad-2f5c1bb5974e" name="Mortgage Payments Calculator"/>
9       <view id="f01d86a6-89ec-11e3-a04c-7f2aac50fa01" name="Puget Sound Home Sales by Index"/>
10      <view id="f01d86a6-89ec-11e3-8c16-9bae1570d152" name="Mercer Island Home Sales"/>
11    </views>
12  </tsResponse>
13

```

Description	Return all the views for the specified workbook, optionally including usage statistics.
Method and URI	<p>GET /api/2.0/sites/site-id/workbooks/workbook-id/views</p> <p>GET /api/2.0/sites/site-id/workbooks/workbook-id/views?includeUsageStatistics=false</p>
Message Payload	None
Response Code	200
Response Payload	<tsResponse> <views> <view id="view1-id" name="view1-name" /> <view id="view2-id" name="view2-name" /> ... more views ... </views> </tsResponse>

Query Views with Usage Statistics

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) in which your view lives
- Use the **Site ID** to list Users in the site (via [Get Users On Site](#))
- Pick a **User ID** and feed it to [Query Workbooks For User](#) in order to get a list of workbooks owned by this user
- Grab the **Workbook ID** and pass [Query Views for a Workbook](#)

The screenshot shows a REST API client interface. At the top, there's a header bar with the title "WORKBOOK management: Query Views for a Workbook with Usage Statistics". Below it, a sub-header says "Get a list of views in a Workbook, including view counts". The URL field contains the endpoint: "ie7-49f2-8624-e77426bbd490/workbooks/efbbe45a-89ec-11e3-8e3a-1bae9933f8ef/views?includeUsageStatistics=true". A red arrow points to the "Workbook ID" part of the URL. To the right of the URL is a "Method" dropdown set to "GET", and buttons for "URL params" and "Headers (1)". Below the URL, there are two "X-Tableau-Auth" fields, one labeled "Auth Token" with value "be96cb5e71e471397b38443aea16a457" and another labeled "Site ID". There are also "Header" and "Value" input fields. Below these are buttons for "Send", "Save", "Preview", "Add to collection", and "Reset". Underneath the main form, there's a status bar showing "Body Headers (14)", "STATUS 200 OK", and "TIME 97 ms". At the bottom, there are tabs for "Pretty", "Raw", "Preview", "JSON", and "XML". The "Raw" tab is selected, displaying the XML response. The XML content is as follows:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse
3   xmlns="http://tableausoftware.com/api"
4   xmlns: xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
      http://tableausoftware.com/api/ts-api-2.0.xsd">
5     <views>
6       <view id="f01d86a6-89ec-11e3-9f4c-17a6772f6520" name="Real Estate Market Report">
7         <usage totalViewCount="0"/>
8       </view>
9       <view id="f01d86a6-89ec-11e3-a876-87188651a4fd" name="Seattle Area Sales Activity">
10      <usage totalViewCount="0"/>
11    </view>
12    <view id="f01d86a6-89ec-11e3-82ad-2f5c1bb5974e" name="Mortgage Payments Calculator">
13      <usage totalViewCount="0"/>
14    </view>
15    <view id="f01d86a6-89ec-11e3-a04c-7f2aac50fa01" name="Puget Sound Home Sales by Index">
16      <usage totalViewCount="1"/>
17    </view>
18    <view id="f01d86a6-89ec-11e3-8c16-9bae1570d152" name="Mercer Island Home Sales">
19      <usage totalViewCount="1"/>
20    </view>
21  </views>
22 </tsResponse>

```

Method and URI	GET /api/2.0/sites/site-id/workbooks/ workbook-id /views?includeUsageStatistics=true
Message Payload	None
Response Code	200
Response Payload	<tsResponse> <views> <view id="view1-id" name="view1-name" >

```

<usage totalViewCount="total-count1" />

</view>

<view id="view2-id" name="view2-name" >

<usage totalViewCount="total-count2" />

</view>

... more views ...

</views>

</tsResponse>

```

Business Logic and Error Conditions

Non-Administrator User Access

Tableau Server users can access this API method, but they can only query workbook views for which they have the **Read** capability allowed (either explicitly or implicitly).

Error Conditions

HTTP Response Code	Condition	Details
400	Invalid parameter value	The <code>includeUsageStatistics</code> was provided with a value other than <code>true</code> .
403	Read forbidden	A Tableau Server user invoked the API method and attempted to access a resource for which they do not have the Read capability.
404	Site not found	The site ID given in the URI does not correspond to an existing site.
404	Workbook not found	The workbook ID given in the payload does not correspond to an existing workbook.
405	Invalid request method	Request type was something other than a GET.

Query Workbook Preview Image

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site (via [Query Site](#)) in which your workbook lives
- Use the **Site ID** to list Users in the site (via [Get Users On Site](#))
- Pick a **User ID** and feed it to [Query Workbooks For User](#) in order to get a list of workbooks owned by this user
- Grab the **Workbook ID** and pass [Query Workbook Preview Image](#)

The screenshot shows two tools side-by-side. On the left is the Postman interface, which has a red box around the URL field containing `2.0/sites/94f96a34-64e7-49f2-8624-e77426bb490/workbooks/efbbe45a-89ec-11e3-8e3a-1bae9933f8ef/previewImage`. To its right is the Charles proxy tool, which also has a red box around the same URL. Both show a successful `200 OK` response with a `176 ms` time. A note in the Charles interface says: "Postman doesn't display images as a response, so use Web Proxy to inspect the response from Tableau." Below the tools is a table comparing various API parameters.

Description	Returns the thumbnail for a specified workbook. Usually the image that's returned is for the first sheet in the workbook.
Method and URI	GET /api/2.0/sites/site-id/workbooks/workbook-id/previewImage
Message Payload	None
Response Code	200
Response Payload	The workbook's preview image in PNG format (MIME media type image/png).

Business Logic and Error Conditions

For on-premise Tableau Server deployments and for Tableau Public, the system returns the preview image associated with the lowest index view in the workbook.

Non-Administrator User Access

Tableau Server users can use this API method, but they can only fetch preview images for workbooks for which they have the **Read** capability allowed (either explicitly or implicitly).

Error Conditions

HTTP Response Code	Condition	Details

403	Read forbidden	A Tableau Server user invoked the API method and attempted to query workbook views for which he or she did not have the Read capability allowed.
404	Site not found	The site ID given in the URI does not correspond to an existing site.
404	Workbook not found	The workbook ID given in the payload does not correspond to an existing workbook.
405	Invalid request method	Request type was something other than a GET.

Query Workbooks for User

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** in which the user lives (use [Query Site](#))
- Use the **Site ID** to list users in the site (via [Get Users On Site](#))
- Pull the **User ID** that you need from the **Get Users On Site** response
- Feed the **Site ID** and **User ID** to the URI

User management: Query workbooks for a user

Get a list of workbooks owned by a specific user **Uri, including Site & User ID**

Method **GET**

5/api/2.0/sites/94f96a34-64e7-49f2-8624-e77426bbd490/users/12d0eb42-1112-4df5-b60d-5dc18aaef3294/workbooks

X-Tableau-Auth **Auth Token** 8987dee32b1e0e770c663f481c87680a

Header Value

Send Save Preview Add to collection Reset

Body Headers (15) STATUS 200 OK TIME 271 ms

Pretty Raw Preview JSON XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
      http://tableausoftware.com/api/ts-api-2.0.xsd">
5     <pagination pageNumber="1" pageSize="100" totalAvailable="2"/>
6     <workbooks>
7       <workbook id="efbbe45a-89ec-11e3-8bae9933f8ef" name="Real Estate"> Workbook ID
8         <project id="efbbe45a-89ec-11e3-8cc3-8cc3c7a01e1f7f4d" name="Tableau Samples"/>
9         <tags></tags>
10        <views>
11          <view id="f01d86a6-89ec-11e3-9f4c-17a6772f6520"/>
12          <view id="f01d86a6-89ec-11e3-a876-87188651a4fd"/>
13          <view id="f01d86a6-89ec-11e3-82ad-2f5c1bb5974e"/>
14          <view id="f01d86a6-89ec-11e3-a04c-7f2aac50fa01"/>
15          <view id="f01d86a6-89ec-11e3-8c16-9bae1570d152"/>
16        </views>
17      </workbook>
18      <workbook id="efbbe45a-89ec-11e3-869b-0345eb19b813" name="Business Dashboard">
19        <project id="efafffc44-89ec-11e3-8cc9-c7a81e1f7f4d" name="Tableau Samples"/>
20        <tags></tags>
21        <views>
22          <view id="f0166268-89ec-11e3-9218-831d5485be7a"/>
23          <view id="f01b244c-89ec-11e3-b785-37346853e4b5"/>
24          <view id="f01d86a6-89ec-11e3-b37b-274756f69911"/>
25          <view id="f01d86a6-89ec-11e3-93db-a315e237529e"/>
26        </views>
27      </workbook>
28    </workbooks>
29  </tsResponse>
```

Description	Return the workbooks associated with a specific user, optionally including paging parameters.
Method and URI	<p>GET /api/2.0/sites/site-id/users/user-id/workbooks</p> <p>GET /api/2.0/sites/site-id/users/user-id/workbooks?pageSize=page-size&pageNumber=page-number</p>
Message Payload	None
Response Code	200
Response Payload	<tsResponse> <pagination pageNumber="pageNumber" pageSize="page-size" totalAvailable="total-available" /> <workbooks> <workbook id="workbook1-id">

```
        name="some-name" >  
  
        <project id="project-id" name="project-name" />  
  
        <tags>  
            <tag label="tag1"/>  
            <tag label="tag2"/>  
            ... more tags ...  
        </tags>  
  
        <views>  
            <view id="view1-id"/>  
            <view id="view2-id"/>  
            ... more views ...  
        </views>  
  
    </workbook>  
  
    <workbook id="workbook2-id"  
        name="some-name" >  
        <project id="project-id" name="project-name" />  
  
        <tags>  
            <tag label="tag1"/>  
            <tag label="tag2"/>  
            ... more tags ...  
        </tags>  
  
        <views>  
            <view id="view1-id"/>  
            <view id="view2-id"/>  
            ... more views ...  
        </views>  
  
    </workbook>  
  
    </workbooks>  
  
    ... more workbooks ...  
  
</tsResponse>
```

Business Logic and Error Conditions

Non-Administrator User Access

Tableau Server users can use this API method, but they can only query workbooks for which they have the **Read** capability allowed (either explicitly or implicitly).

Pagination Parameter Defaults

By default the page size is 100 and page 1 is served. One or both parameters can be overridden explicitly.

Error Conditions

HTTP Response Code	Condition	Details
403	Read forbidden	A Tableau Server user invoked the API method and attempted to query workbook views for which he or she did not have the Read capability allowed.
404	Site not found	The site ID given in the URI does not correspond to an existing site.
404	User not found	The user ID given in the URI does not correspond to an existing user.
405	Invalid request method	Request type was something other than a GET.

Add User to Site

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site to which you want to add the user (use [Query Site](#))
- Pass the Site ID to **Add User to Site**

User Management: Add User to Site

Add user to a site Uri, including Site ID

Method

POST POST URL params Headers (1)

X-Tableau-Auth Auth Token

808cb958dfcb30837c16bf8adc5b2f80

Manage presets

Header Value

form-data x-www-form-urlencoded raw Text

```

1 <tsRequest>
2   <user name="RESTUser"
3     role="Interactor"
4     public="false"
5     contentAdmin="false"
6     suppressGettingStarted="true"/>
7 </tsRequest>

```

Username, licensing information and admin status

Send Save Preview Add to collection Reset

Body Headers (14) STATUS 201 Created TIME 499 ms

Pretty Raw Preview JSON XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse>
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
5   http://tableausoftware.com/api/ts-api-2.0.xsd">
6     <user id="8ca2f254-c3a0-4d96-8bd8-3210f66c268c" name="RESTUser" role="Interactor" publish="false" contentAdmin="false"
7     suppressGettingStarted="true"/>

```

Description	Adds a user to the specified site.
Method and URI	POST /api/2.0/sites/ site-id /users/
Message Payload	<tsRequest> <user name="user-name" role="some-role" publish="true-or-false" contentAdmin="true-or-false*" suppressGettingStarted="true-or-false" /> </tsRequest>
Response Code	201
Response Payload	<tsResponse> <user id="user-id" name="user-name"

	<pre> role="some-role" publish="true-or-false" contentAdmin="true-or-false" suppressGettingStarted="true-or-false" /> </tsResponse></pre>
Response Headers	Location: /api/2.0/sites/site-id/users/new-user-id

Payload Details

There are three valid roles available: Interactor, Viewer, and Unlicensed.

The `contentAdmin` parameter can only be true if the role is Interactor. For any other role, a value of `contentAdmin` in the request payload will be ignored, and the value `false` will be used and subsequently returned in the response payload.

The `suppressGettingStarted` parameter may be omitted. If it is, the user will retain their existing preference if he or she is already a member of another site. If the user is new to the system, the preference will be set to the default according to user type: enabled for site administrators, suppressed for others.

Error Conditions

HTTP Response Code	Condition	Details
404	Site not found	The site ID given in the URI does not correspond to an existing site.
404	User name not found	The user name given in the payload does not correspond to an existing user in the system.
405	Invalid request method	Request type was something other than a <code>POST</code> or a <code>GET</code> .
409	User conflict	The given user is already registered on the site.
409	Guest user conflict	The Tableau Online API disallows adding a user with the guest role to a site.

Get Users on Site

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site for which you want to list users (use [Query Site](#))

- Pass the Site ID to Get Users On Site

User Management: Get Users on Site

List all the users in a specific site

Uri including Site ID

Method

http://10.211.55.4/api/2.0/sites/c174d512-a1fd-4cccd-b750-2f67b1cb877c/users/

GET

X-Tableau-Auth Auth Token 808cb958dfcb30837c16bf8adc5b2f80

Header Value

Manage presets

Send Save Preview Add to collection Reset

Body Headers (15) STATUS 200 OK TIME 172 ms

Pretty Raw Preview JSON XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
http://tableausoftware.com/api/ts-api-2.0.xsd">
5     <users>
6       <user id="4f67be3e-5338-45f4-b578-f19446cd636f" name="DanAdmin" role="Interactor" publish="false" contentAdmin="false" lastLogin=""
externalAuthUserId="" />
7       <user id="e04d08d7-4455-4975-9662-3fe1a013df74" name="Paul" role="Interactor" publish="false" contentAdmin="false" lastLogin=""
externalAuthUserId="" />
8       <user id="8ca2f254-c3a0-4d96-8bd8-3210f66c268c" name="RESTUser" role="Interactor" publish="false" contentAdmin="false" lastLogin=""
externalAuthUserId="" />
9       <user id="feba71c8-5b08-4a17-9819-5c3081d2e7f7" name="russch" role="Interactor" publish="false" contentAdmin="false" lastLogin="2014-06-
01T12:47:07Z" externalAuthUserId="" />
10    </users>
11  </tsResponse>
12

```

2	Returns the users associated with the specified site.
Method and URI	GET /api/2.0/sites/ site-id /users/
Message Payload	None
Response Code	200
Response Payload	<tsResponse> <users> <user id="user1-id" name="user1-name" role="some-role" publish="true-or-false" contentAdmin="true-or-false" lastLogin="YYYY-MM-DDTHH:MM:SSZ" externalAuthUserId="authentication-id-from-external-provider"/>

```

<user id="user2-id"
      name="user2-name"
      role="some-role"
      publish="true-or-false"
      contentAdmin="true-or-false"
      lastLogin="YYYY-MM-DDTHH:MM:SSZ"
      externalAuthUserId="authentication-id-from-external-provider" />
...
</users>
</tsResponse>

```

Error Conditions

HTTP Response Code	Condition	Details
404	Site not found	The site ID given in the URI does not correspond to an existing site.
405	Invalid request method	Request type was something other than a POST or a GET.

Remove User from Site

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Get the **Site ID** of the site from which you want to delete the user (use [Query Site](#))
- Get the **User ID** of the user you wish to delete via [Get Users On Site](#)
- Pass the **User ID** to **Remove User from Site**

The screenshot shows the Postman interface with the following details:

- User Management: Remove User from Site**
- Remove a specific user from a site**
- Uri, including Site ID and User ID**: http://10.211.55.4/api/2.0/sites/c174d512-a1fd-4cc0-b750-2f67b1cb877c/users/8ca2f254-c3a0-4d96-8bd8-3210f66c2f
- Method**: DELETE
- X-Tableau-Auth Auth Token**: 808cb958dfcb30837c16bf8adc5b2f80
- Header**: Value
- Content Type**: Text
- Send** button is highlighted.
- Response Headers**: Headers (12) - STATUS: 204 No Content, TIME: 400 ms
- Response Body**: SUCCESS

Description	Removes a user from the specified site.
Method and URI	DELETE /api/2.0/sites/site-id/users/ user-id
Message Payload	None
Response Code	204
Response Payload	None

Business Logic and Error Conditions

If a user still owns content on Tableau Server, the user cannot be deleted. The ownership of the user's content will need to move to a different user first.

Error Conditions

HTTP Response Code	Condition	Details
400	Deletion failed	Some other problem arose that prevented the user from being removed from the site.
404	Site not found	The site ID given in the URI does not correspond to an existing site.
404	User not found	The user ID given in the URI does not correspond to an existing user.
405	Invalid request method	Request type was something other than a DELETE, a POST, or a GET.
409	User asset conflict	The given user still owns content on Tableau Server.

Create Site

What you need:

- Sign In to get a X-Tableau-Auth token
- Call Create Site

The screenshot shows the Postman application interface for making an API call. The 'Sites: Create Site' collection is selected. The 'Create a new site' request is being configured.

- Uri:** http://10.211.55.4/api/2.0/sites (highlighted with a red box)
- Method:** POST (highlighted with a red box)
- Headers:** X-Tableau-Auth (Auth Token) - 808cb958dfcb30837c16bf8adc5b2f80 (highlighted with a red box)
- Body:** Text (highlighted with a red box)
The body contains XML payload:

```
1 <tsRequest>
2   <site name="A Random Site"
3     contentUrl="Site7"
4     adminMode="ContentAndUsers"
5     userQuota="10"
6     storageQuota="40"
7     disableSubscriptions="false"/>
8 </tsRequest>
```
- Buttons:** Send (blue), Save, Preview, Add to collection, Reset
- Response:** Headers (14), STATUS 201 Created, TIME 335 ms
Pretty, Raw, Preview, JSON, XML
- Raw Response:**

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
5   http://tableausoftware.com/api/ts-api-2.0.xsd">
6   <site id="4466751d-29e1-483f-9c28-a4e07870d16a" name="A Random Site" contentUrl="Site7" adminMode="ContentAndUsers" userQuota="10"
7   storageQuota="40" disableSubscriptions="false" state="Active" statusReason=""></site>
8 </tsResponse>
```

Description	Creates a site on Tableau Server with the given metadata.
Method and URI	POST /api/2.0/sites/
Message Payload	<tsRequest> <site name="site-name" contentUrl="some-content-url"

	<pre> adminMode="some-admin-mode" userQuota="num-users" storageQuota="limit-in-megabytes" disableSubscriptions="false" /> </tsRequest></pre>
Response Code	201
Response Payload	<pre> <tsResponse> <site id="new-site-id" name="site-name" contentUrl="the-content-url" adminMode="the-admin-mode" userQuota="num-users" storageQuota="limit-in-megabytes" disableSubscriptions="false" /> </tsResponse></pre>
Response Headers	Location: /api/2.0/sites/new-site-id

Payload Details

At the time this section was edited, there were two valid admin modes available: ContentOnly, and ContentAndUsers. Note that `adminMode` is not a required parameter.

Error Conditions

HTTP Response Code	Condition	Details
405	Invalid request method	Request type was something other than a POST or a GET.
409	Site name conflict	The site name in the request already belongs to an existing site in the system.
409	Site URL conflict	The content URL in the request already belongs to an existing site in the system.

Update Site

What you need:

- Sign In to get a X-Tableau-Auth token
- Use [Query Sites](#) to get a list of the Site IDs on your Server
- Choose the Site ID for the site you wish to update and pass to Update Site

The screenshot shows a cURL-like interface for making API requests. The URL field contains `http://10.211.55./api/2.0/sites/4466751d-29e1-483f-9c28-a4e07070d16a`. The Method dropdown is set to `PUT`. The `X-Tableau-Auth` header is set to `Auth Token 808cb958dfcb30837c16bf8adc5b2f80`. The body of the request contains XML code to update the site name:

```
1 <tsRequest>
2   <site name="Updated Site Name" <!-- Updated Site Name -->
3     storageQuota="40"
4     disableSubscriptions="false"/>
5 </tsRequest>
```

Below the request area, the response status is shown as `200 OK` with a `TIME` of `225 ms`. The response body is displayed in XML format:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse>
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
http://tableausoftware.com/api/ts-api-2.0.xsd">
5     <site id="4466751d-29e1-483f-9c28-a4e07070d16a" name="Updated Site Name" contentUrl="Site7" adminMode="ContentAndUsers" userQuota="10"
storageQuota="40" disableSubscriptions="false" state="Active" statusReason=""></site>
6   </tsResponse>
7 
```

Description	Modifies the metadata of the specified site.
Method and URI	PUT /api/2.0/sites/ site-id
Message Payload	<tsResponse> <site name="some-name" contentUrl="some-content-url" adminMode="some-admin-mode" userQuota="num-users" state="active-or-suspended">

	<pre> statusReason="reason-for-state" storageQuota="limit-in-megabytes" disableSubscriptions="true-or-false" /> </tsResponse></pre>
Response Code	200
Response Payload	<pre> <tsResponse> <site name="some-name" contentUrl="some-content-url" adminMode="some-admin-mode" userQuota="num-users" state="active-or-suspended" statusReason="reason-for-state" storageQuota="limit-in-megabytes" disableSubscriptions="true-or-false" /> </tsResponse></pre>

Payload Details

At the time this section was edited, there were two valid admin modes available: `ContentOnly`, and `ContentAndUsers`.

Business Logic and Error Conditions

Payload Variations

Any combination of the attributes inside the `<site>` element is valid. Only those attributes present will result in updates to their values in the site. If no attributes are present, then the update will not take effect.

Error Conditions

HTTP Response Code	Condition	Details
400	Invalid admin mode	An admin mode parameter was provided in the request with an invalid value.
400	Invalid state	A state parameter was provided in the request with an invalid value

404	Site not found	The site ID given in the URI does not correspond to an existing site.
404	User name not found	The user name given in the payload does not correspond to an existing user in the system.
405	Invalid request method	Request type was something other than a PUT, a DELETE, or a GET.
409	Site name conflict	The new site name in the request already belongs to an existing site in the system.
409	Site URL conflict	The new content URL in the request already belongs to an existing site in the system.

Query Site

What you need:

- Sign In to get a X-Tableau-Auth token

The screenshot shows the Postman interface with the following details:

- Method:** GET
- Uri, site name parameter:** http://10.211.55.4/api/2.0/sites/russell?key=name
- Auth Token:** 808cb958dfcb30837c16bf8adc5b2f80
- Headers:** X-Tableau-Auth
- Status:** 200 OK
- Body:** XML response containing site metadata (id, name, contentUrl, adminMode, userQuota, storageQuota, disableSubscriptions, state, statusReason).

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse
3   xmlns="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
http://tableausoftware.com/api/ts-api-2.0.xsd">
4     <site id="c174d512-a1fd-4cc4-b750-2f67b1cb877c" name="russell" contentUrl="russellSite" adminMode="ContentAndUsers" userQuota="11"
storageQuota="40" disableSubscriptions="false" state="Active" statusReason=""></site>
5   </tsResponse>
  
```

Description	Returns metadata on the specified site, which can be queried by name, site-id, or the site URL.
Method and URI	Query by Site ID GET/api/2.0/sites/ site-id

	Query by Site Name GET/api/2.0/sites/ site-name ?key=name
	Query by URL Namespace GET/api/2.0/sites/ site-url-namespace ?key=contentUrl
Message Payload	None
Response Code	200
Response Payload	<pre> <tsResponse> <site id="site-id" name="some-name" contentUrl="some-content-url" adminMode="some-admin-mode" userQuota="num-users" storageQuota="limit-in-megabytes" state="active-or-suspended" statusReason="reason-for-state" /> </tsResponse> </pre>

Site Information Including Storage Usage

What you need:

- **Sign In** to get a **X-Tableau-Auth** token

Sites: Site Information Including Storage

View information (including storage info) about a site - similar to Query Site but with more stuff.

Method

GET URL params Headers (1)

URI, including includeUsage parameter

Send Save Preview Add to collection Reset

Auth Token 808cb958dfcb30837c16bf8adc5b2f80

Header Value

Body Headers (15) STATUS 200 OK TIME 96 ms

Pretty Raw Preview JSON XML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tsResponse
3   xmlns="http://tableausoftware.com/api"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://tableausoftware.com/api
  http://tableausoftware.com/api/ts-api-2.0.xsd">
5     <site id="4466751d-29e1-483f-9c28-a4e07070d16a" name="Updated Site Name" contentUrl="Site7" adminMode="ContentAndUsers" userQuota="10"
6       storageQuota="40" disableSubscriptions="false" state="Active" statusReason="">
7       <usage numUsers="0" storage="0"/> Usage numbers
8   </site>
9 </tsResponse>

```

Method and URI	<p>Query by Site ID GET/api/2.0/sites/site-id?includeUsage=true</p> <p>Query by Name GET/api/2.0/sites/site-name?key=name&includeUsage=true</p> <p>Query by URL Namespace GET/api/2.0/sites/site-url-namespace?key=contentUrl&includeUsage=true</p>
Message Payload	None
Response Code	200
Response Payload	<pre> <tsResponse> <site id="site-id" name="some-name" contentUrl="some-content-url" adminMode="some-admin-mode" userQuota="num-users" storageQuota="limit-in-megabytes" state="active-or-suspended" </pre>

```

    statusReason="reason-for-state" >

    <usage numUsers="the-number-of-users"
           storage="the-storage-in-megabytes"
    </usage>
    </site>
</tsResponse>

```

Error Conditions

HTTP Response Code	Condition	Details
404	Site not found	The site ID or URL namespace given in the URI does not correspond to an existing site.
405	Invalid request method	Request type was something other than a <code>GET</code> , <code>DELETE</code> , or <code>PUT</code> .

Delete Site

What you need:

- **Sign In** to get a **X-Tableau-Auth** token
- Use [Query Sites](#) to get a list of the Site IDs on your Server
- Choose the **Site ID** for the site you wish to update and pass to **Delete Site**

Description	Deletes the specified site.
Method and URI	Delete by Site ID <code>DELETE/api/2.0/sites/site-id</code>
	Delete by Site Name <code>DELETE/api/2.0/sites/site-name?key=name</code>
	Delete by URL Namespace <code>DELETE/api/2.0/sites/site-url-namespace?key=contentUrl</code>
Message Payload	None
Response Code	204

Response Payload	None
------------------	------

Error Conditions

HTTP Response Code	Condition	Details
403	Deletion forbidden	Attempt to delete the default Tableau Server site.
404	Site not found	The site ID or URL namespace given in the URI does not correspond to an existing site.
405	Invalid request method	Request type was something other than a GET, DELETE, or PUT.