<http://rest.testsparker.com/docs/>

# Netsparker Vulnerable REST API

## Netsparker Vulnerable REST API Test Web Application Documentation

## General

|  |  |
| --- | --- |
| Technologies: | Ubuntu 18, Apache, PHP 7.1, MySQL |
| Supported Formats: | JSON, XML |
| Supported Authentication Types: | JSON Web Token, Basic Authentication, OAuth2 |
| Supported Importable File Formats: | Swagger/OpenAPI, Postman, Fiddler |

## How to scan REST API?

#### Auto Generated Scan Profile by Netsparker Assistant

1. Open Netsparker Standard.
2. From the **Home** tab, click **New**. The **Start a New Website or Web Service URL** dialog is displayed.
3. /In the **Target Website** field, paste one of these URLs:
   * For Basic Authentication => http://rest.testsparker.com/basic\_authentication/api/
   * For JSON Web Token => http://rest.testsparker.com/jwt/api/
   * For OAuth2 => http://rest.testsparker.com/oauth/api/
4. Click **Start Scan**.

**Netsparker Assistant** automatically imports the required configurations.

#### Manual Scan

1. Download the [Scan Profile and Policy files](http://rest.testsparker.com/docs/#api-GettingStarted-6).
   * Copy the scan profile file to the following directory on your computer: C:\Users\{username}\Documents\Netsparker\Profiles
   * Copy the scan policy file to under this directory on your computer: C:\Users\{username}\Documents\Netsparker\Policies
2. Open Netsparker Standard.
3. From the **Home** tab, click **New**.
4. From the **Target Website or Web Service URL** panel, click **Previous Settings**, and select the relevant Scan Profile.
5. Click **Start Scan**.

**Note:** Don't forget to review your Imported Linked, URL Rewrites and Authentication changes, so that you can see how they will affect the scan.

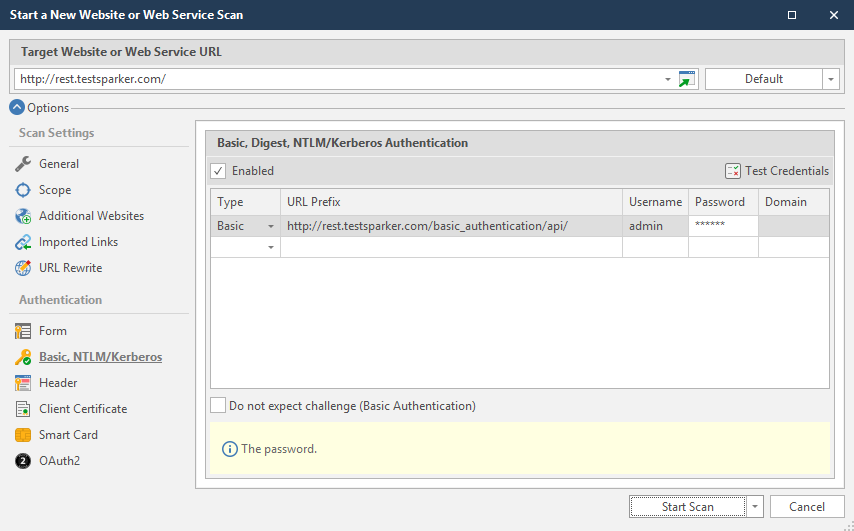
## Authentication Types

#### 1. Basic Authentication

Netsparker supports Basic Authentication. You need to set the following configurations in the **Basic, NTLM/Kerberos** tab in the **Authentication** section of the **Start New Website or Web Service Scan** dialog.

Credentials for the Netsparker Vulnerable REST API:

* Type: Basic
* URL Prefix: http://rest.testsparker.com/basic\_authentication/api/
* Username: admin
* Password: 123456



Netsparker will add this HTTP header to each request: Authorization: Basic YWRtaW46MTIzNDU2

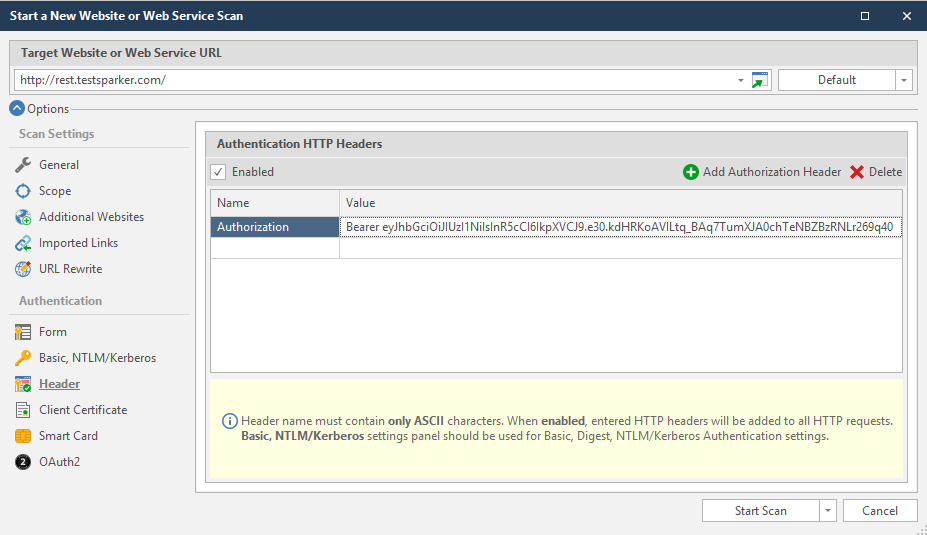
#### 2. JSON Web Token (JWT)

Netsparker supports JSON Web Tokens. You can set following configurations in the **Header** tab of the **Authentication** section of the Start New Scan > Authentication >\*\* dialog.

The Netsparker Vulnerable REST API's secret JWT key is: n3tsp4rk3r\_s3cr3t\_jwt\_t0k3n . For further information on how to use the JWT, see [the JTW.io website](https://jwt.io/).: <https://jwt.io/>

Credentials for Netsparker's Vulnerable REST API:

* Name: Authorization
* Value: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40



Netsparker will add this HTTP header to each request: Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40

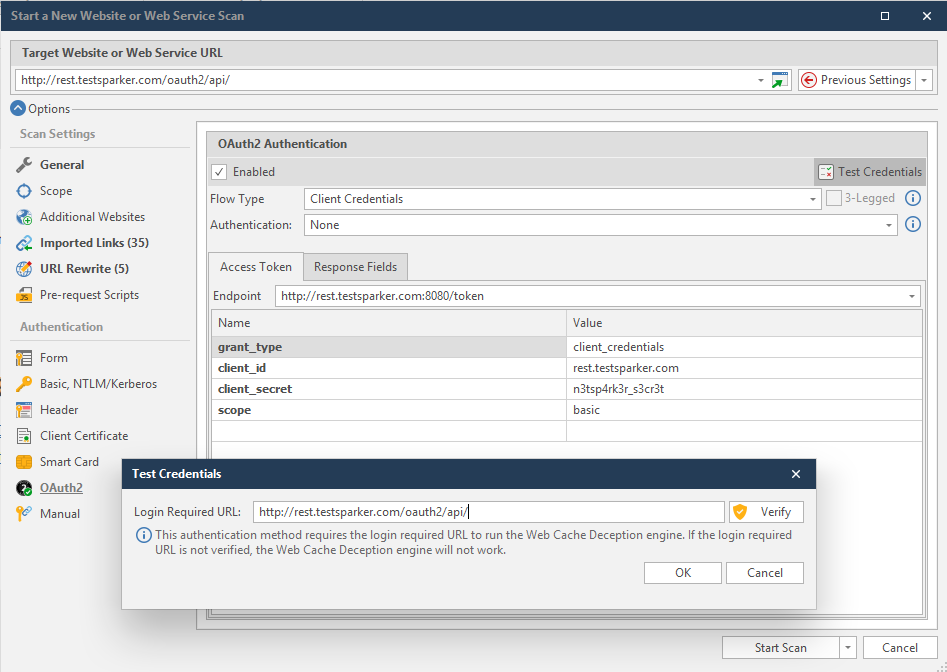
#### 3. OAuth2.0

Netsparker supports OAuth2. You need to set the following configurations in the **OAuth2** tab in the **Authentication** section of the **Start New Website or Web Service Scan** dialog.

Credentials for the Netsparker Vulnerable REST API:

* Flow Type: Client Credentials
* Authentication: None
* Access Token:
  + Endpoint: http://rest.testsparker.com:8080/token
  + Content Type: application/x-www-form-urlencoded
  + Method: POST
  + grant\_type: client\_credentials
  + client\_id: rest.testsparker.com
  + client\_secret: n3tsp4rk3r\_s3cr3t
  + scope: basic
* Response Fields
  + Access Token: access\_token
  + Refresh Token: refresh\_token
  + Expire: expires\_in
  + Token Type: token\_type

Click **Test Credentials** button and enter this URL to input: http://rest.testsparker.com/oauth2/api/



Netsparker will add this HTTP header to each request: Authorization: Bearer {access\_token}

## Supported Request Body Formats

#### 1. JSON

If you use the JSON format in the HTTP request body, you should add this header to each request: Content-type: application/json

Sample request:

PUT /basic\_authentication/api/comments/101 HTTP/1.1

Host: rest.testsparker.com

Authorization: Basic YWRtaW46MTIzNDU2

Content-type: application/json

Accept: application/json

{

"user\_id": "21",

"post\_id": "34",

"comment": "test comment"

}

#### 2. XML

If you use the XML format in the HTTP request body, you should add this header to each request: Content-type: application/xml

Sample request:

PUT /basic\_authentication/api/comments/101 HTTP/1.1

Host: rest.testsparker.com

Authorization: Basic YWRtaW46MTIzNDU2

Content-type: application/xml

Accept: application/xml

<?xml version="1.0" encoding="UTF-8" ?>

<comment>

<user\_id>21</user\_id>

<post\_id>34</post\_id>

<comment>test comment</comment>

</comment>

## Accepted Response Body Formats

#### 1. JSON

If you want to get JSON data in the HTTP response body, you should add this header to each request: Accept: application/json

Sample request:

PUT /basic\_authentication/api/comments/101 HTTP/1.1

Host: rest.testsparker.com

Authorization: Basic YWRtaW46MTIzNDU2

Content-type: application/json

Accept: application/json

{

"user\_id": "21",

"post\_id": "34",

"comment": "test comment"

}

Sample response:

HTTP/1.1 200 OK

Content-Type: application/json

{

"comment":{

"user\_id":"21",

"post\_id":"34",

"comment":"test comment",

"id":"101"

},

"response":false

}

#### 2. XML

If you want to get XML data in the HTTP response body, you should add this header to each request: Accept: application/xml

Sample request:

PUT /basic\_authentication/api/comments/101 HTTP/1.1

Host: rest.testsparker.com

Authorization: Basic YWRtaW46MTIzNDU2

Content-type: application/xml

Accept: application/xml

<?xml version="1.0" encoding="UTF-8" ?>

<comment>

<user\_id>21</user\_id>

<post\_id>34</post\_id>

<comment>test comment</comment>

</comment>

Sample response:

HTTP/1.1 200 OK

Content-Type: application/xml

<?xml version="1.0" encoding="UTF-8" ?>

<response>

<comment>

<user\_id>21</user\_id>

<post\_id>34</post\_id>

<comment>test comment</comment>

<id>101</id>

</comment>

<response></response>

</response>

## Download the Scan Profile and Scan Policy

|  | **Profile** | **Policy** |
| --- | --- | --- |
| Basic Authentication (JSON + XML body) | [rest.testsparker.com (Basic Auth) Profile.xml](http://rest.testsparker.com/files/rest.testsparker.com%20(Basic%20Auth)%20Profile.xml) | [REST API Policy (rest.testsparker.com).xml](http://rest.testsparker.com/files/REST%20API%20Policy%20(rest.testsparker.com).xml) |
| JSON Web Token (JSON + XML body) | [rest.testsparker.com (JWT) Profile.xml](http://rest.testsparker.com/files/rest.testsparker.com%20(JWT)%20Profile.xml) |
| OAuth (JSON + XML body) | [rest.testsparker.com (OAuth2) Profile.xml](http://rest.testsparker.com/files/rest.testsparker.com%20(OAuth2)%20Profile.xml) |

## Download Importable Links Files

|  |  |
| --- | --- |
| Fiddler (Basic Authentication) (JSON + XML body) | [fiddler\_basicauthentication.saz](http://rest.testsparker.com/files/fiddler_basicauthentication.saz) |
| Fiddler (JSON Web Token) (JSON + XML body) | [fiddler\_jwt.saz](http://rest.testsparker.com/files/fiddler_jwt.saz) |
| Fiddler (OAuth) (JSON + XML body) | [fiddler\_oauth2.saz](http://rest.testsparker.com/files/fiddler_oauth2.saz) |
| Postman (Basic Authentication) (JSON + XML body) | [postman\_basicauthentication.json](http://rest.testsparker.com/files/postman_basicauthentication.json) |
| Postman (JSON Web Token) (JSON + XML body) | [postman\_jwt.json](http://rest.testsparker.com/files/postman_jwt.json) |
| Postman (OAuth) (JSON + XML body) | [postman\_oauth2.json](http://rest.testsparker.com/files/postman_oauth2.json) |
| Swagger (Basic Authentication) (JSON + XML body) | [openapi-swagger\_basicauthentication.yaml](http://rest.testsparker.com/files/openapi-swagger_basicauthentication.yaml) |
| Swagger (JSON Web Token) (JSON + XML body) | [openapi-swagger\_jwt.yaml](http://rest.testsparker.com/files/openapi-swagger_jwt.yaml) |
| Swagger (OAuth) (JSON + XML body) | [openapi-swagger\_oauth2.yaml](http://rest.testsparker.com/files/openapi-swagger_oauth2.yaml) |

# Comment

**GET**

# 1. Get all comments

http://rest.testsparker.com/{auth\_type}/api/comments

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |

**POST**

# 2. Add new comment

http://rest.testsparker.com/{auth\_type}/api/comments

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |

* [JSON Request Body:](http://rest.testsparker.com/docs/#parameter-examples-Comment-AddComment-0_0_0-0)
* [XML Request Body:](http://rest.testsparker.com/docs/#parameter-examples-Comment-AddComment-0_0_0-1)

{

"user\_id": "21",

"post\_id": "34",

"comment": "test comment"

}

**GET**

# 3. Get comment

http://rest.testsparker.com/{auth\_type}/api/comments/{comment\_id}

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |
| comment\_id | Integer | Should be valid comment id. Can see all comments and ids Get all comments endpoint. |

**PUT**

# 4. Edit comment

http://rest.testsparker.com/{auth\_type}/api/comments/{comment\_id}

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |
| comment\_id | Integer | Should be valid comment id. Can see all comments and ids Get all comments endpoint. |

* [JSON Request Body:](http://rest.testsparker.com/docs/#parameter-examples-Comment-EditComment-0_0_0-0)
* [XML Request Body:](http://rest.testsparker.com/docs/#parameter-examples-Comment-EditComment-0_0_0-1)

{

"user\_id": "21",

"post\_id": "34",

"comment": "test comment"

}

**DELETE**

# 6. Delete comment

http://rest.testsparker.com/{auth\_type}/api/comments/{comment\_id}

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |
| comment\_id | Integer | Should be valid comment id. Can see all comments and ids Get all comments endpoint. |

# Post

**GET**

# 1. Get all posts

http://rest.testsparker.com/{auth\_type}/api/posts

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |

**POST**

# 2. Add new post

http://rest.testsparker.com/{auth\_type}/api/posts

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |

* [JSON Request Body:](http://rest.testsparker.com/docs/#parameter-examples-Post-AddPost-0_0_0-0)
* [XML Request Body:](http://rest.testsparker.com/docs/#parameter-examples-Post-AddPost-0_0_0-1)

{

"user\_id": "21",

"title": "test post title",

"content": "test content"

}

**GET**

# 3. Get post

http://rest.testsparker.com/{auth\_type}/api/posts/{post\_id}

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |
| post\_id | Integer | Should be valid post id. Can see all posts and ids Get all posts endpoint. |

**GET**

# 4. Get post comments

http://rest.testsparker.com/{auth\_type}/api/posts/{post\_id}/comments

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |
| post\_id | Integer | Should be valid post id. Can see all posts and ids Get all posts endpoint. |

**PUT**

# 5. Edit post

http://rest.testsparker.com/{auth\_type}/api/posts/{post\_id}

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |
| post\_id | Integer | Should be valid post id. Can see all posts and ids Get all posts endpoint. |

* [JSON Request Body:](http://rest.testsparker.com/docs/#parameter-examples-Post-EditPost-0_0_0-0)
* [XML Request Body:](http://rest.testsparker.com/docs/#parameter-examples-Post-EditPost-0_0_0-1)

{

"user\_id": "21",

"title": "test post title",

"content": "test content"

}

**DELETE**

# 6. Delete post

http://rest.testsparker.com/{auth\_type}/api/posts/{post\_id}

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |
| post\_id | Integer | Should be valid post id. Can see all posts and ids Get all posts endpoint. |

# User

**GET**

# 1. Get all users

http://rest.testsparker.com/{auth\_type}/api/users

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |

**POST**

# 2. Add new user

http://rest.testsparker.com/{auth\_type}/api/users

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |

* [JSON Request Body:](http://rest.testsparker.com/docs/#parameter-examples-User-AddUser-0_0_0-0)
* [XML Request Body:](http://rest.testsparker.com/docs/#parameter-examples-User-AddUser-0_0_0-1)

{

"username": "test\_username",

"email": "test\_user@example.com",

"password": "123456",

"first\_name": "test",

"last\_name": "user"

}

**GET**

# 3. Get user

http://rest.testsparker.com/{auth\_type}/api/users/{username}

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |
| username | String | Should be valid any username. Can see all username Get all users endpoint. |

**GET**

# 4. Get user blogposts

http://rest.testsparker.com/{auth\_type}/api/users/{username}/posts

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |
| username | String | Should be valid any username. Can see all username Get all users endpoint. |

**PUT**

# 5. Edit user

http://rest.testsparker.com/{auth\_type}/api/users/{username}

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |
| username | String | Should be valid any username. Can see all username Get all users endpoint. |

* [JSON Request Body:](http://rest.testsparker.com/docs/#parameter-examples-User-EditUser-0_0_0-0)
* [XML Request Body:](http://rest.testsparker.com/docs/#parameter-examples-User-EditUser-0_0_0-1)

{

"username": "test\_username",

"email": "test\_user@example.com",

"password": "123456",

"first\_name": "test",

"last\_name": "user"

}

**DELETE**

# 6. Delete user

http://rest.testsparker.com/{auth\_type}/api/users/{username}

## Header

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Authorization | String | Allowed values: "Basic YWRtaW46MTIzNDU2", "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.e30.kdHRKoAVILtq\_BAq7TumXJA0chTeNBZBzRNLr269q40" |

## Parameter

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| auth\_type | String | Allowed values: "basic\_authentication", "jwt" |
| username | String | Should be valid any username. Can see all username Get all users endpoint. |