­­­­­­­­­­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**WiKeD User Management Documentation Template**

**Release 1.0.**



Adam Wheeler

01/11/2021

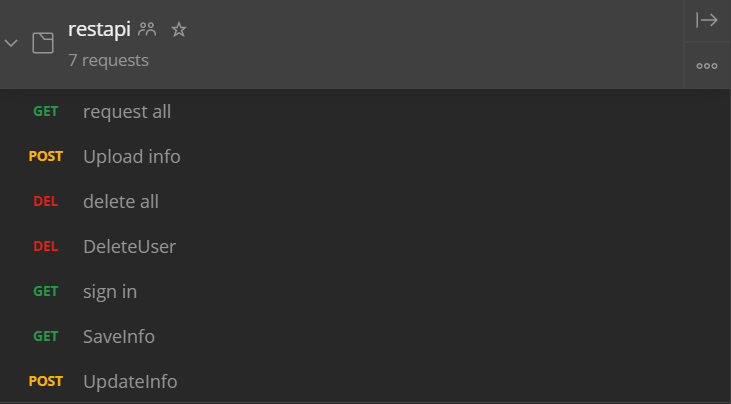
Contents

[Use Case 3](#_Toc86755837)

[Features 4](#_Toc86755838)

[Errors 7](#_Toc86755839)

# Use Case

The ideal use for this software is User management, creating, deleting and storing information in a safe manor.

Above are all the functions available to the admin, some features are only available to the admin.

WiKeD Rest API can be used when creating a website with a sign in page, the API can store the new user’s information such as passwords, encrypted with the MD5 hash algorithm ensuring a secure and safe transfer of information. However, this means that once the password is given the user cannot view their password and will not be able to update the information unless the correct password is given.

# Features

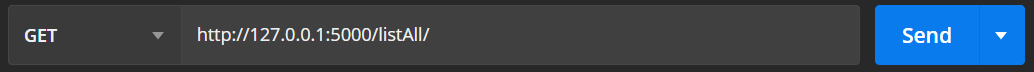
Text

Description automatically generated

Base URL -

As seen above the features of this API are the following

**Request All:** This will return all the users that are currently saved on the system, at the beginning only the admins login is saved. To view all the users

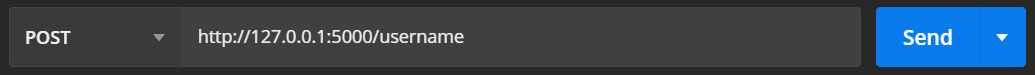
**URL:**

The following request will return the following if no users have been added.

Once more users have been added the list will be updated when the request is sent.

**Add User:** This will add a user to the list if all the requirements are met.

To add a user. This URL is required along with the information of the user.

**URL:**

 **Information of the user:**

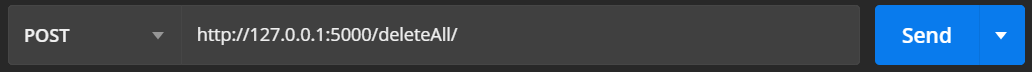
The information must be written this way otherwise the API will return an error to check the documentation.

Prerequisites*: The username must not already be in use.*

*The username must not be over 15 lowercase characters or have any special characters and numbers.*

If all Prerequisites are met a successful add user, will return:

**Delete all:** This will delete all users in the list that have no been saved, this does not include the admin, however you must “be” the admin in the additional information to delete the contents.

**URL**

**Information of the user:**

If this input is given, the entire list that hasn’t been saved is deleted.

Before

[{"id": 1, "username": "admin", "password": "5beeda424743c7f363e567660764a030"}

{"username": "username", "password":"97667d4bc87719fbdf674c65221fcbcf", "id": 2}]

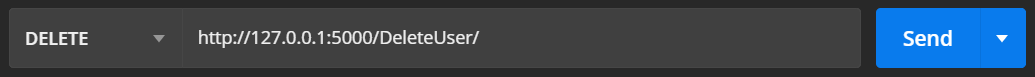
After



**Delete User:** This will delete a specific user if this correct ***ID*** is given.

**ID: the id is added when a user is made and can be found when you request all the information in the list, in the example below it can be found at the end: “id”: 2. The admin account cannot be deleted, it will return an error!**

{"username": "username", "password":"97667d4bc87719fbdf674c65221fcbcf", "id": 2}]

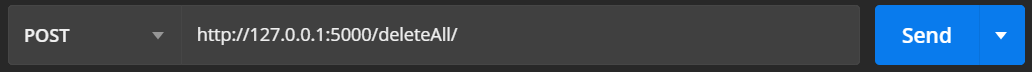
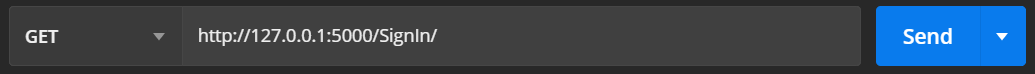
**URL**

**Information of the user:**

The following information would delete the user with the ID of “2”.

Once the user is deleted the entire table the entire table gets a new id value, so its good practise to check the id value before deleting anything and save the table before deleting anything as the information cannot be recovered unless saved.

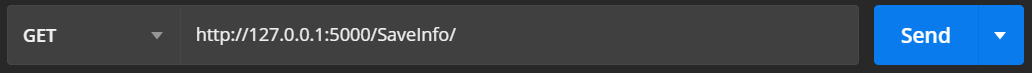
**Sign In:** This feature allows the user to sign in, if the username and password is saved, and then is re-entered, the use will be able to sign into your website with some additional code on the website end.

**URL**

**Information of the user:**

 **Correct user information Incorrect user information**

**Save Info:** This allows for the users that have been recently added to the list to be saved in file, so any information added isn’t lost.

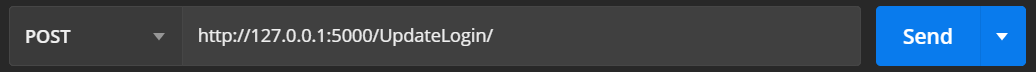
**URL**

**Information of the user:**

Prerequisites*:**you must be signed in with the admin username and password.*

Once the information is saved if the Rest API is restarted the information at the start will be whatever was last saved. This is mainly used If something was deleted that shouldn’t have been.

**Update Info:** This allows the use to change their password if they want to

**URL** 

**Information of the user:**

The additonional information should be the same as the sign in however with a new variable called “newpassword” this is what the new password will be changed to, if the username and password provided are correct the user will be updated. This has to be saved to ensure the change.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Errors

***Syntax***

If you get this error message to check the documentation this means that the additional information entered is incorrectly entered, this might be syntax or an error on my end.

***404 not found***

If you get this error the URL that was entered is incorrect and you should check to see what the URL given for each of the features is.

***500 Internal Server Error***

This error means that the variable name in the additional information is incorrect, it might be spelling or order.

*if you get an error and aren’t sure why, you can email* [*Wheeler\_adam@hotmail.com*](mailto:Wheeler_adam@hotmail.com) *and I will try respond to how to fix it.*