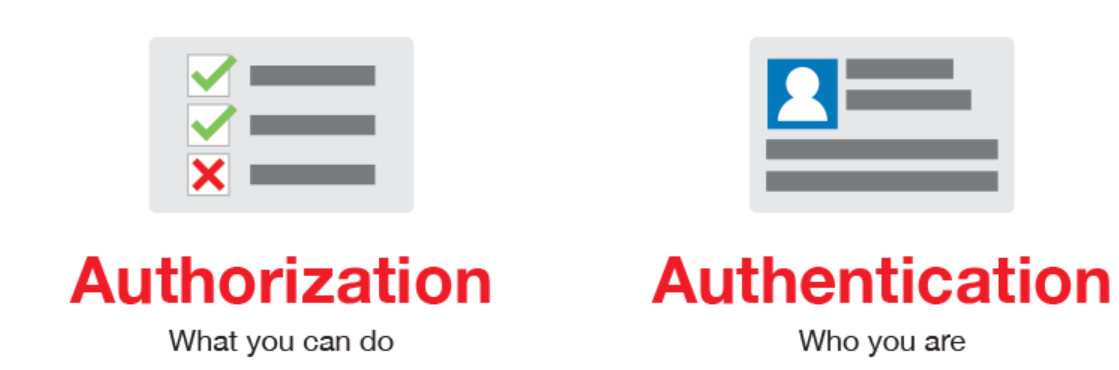
**Request Authentication/ Authorization**



**Authentication:**  gives those users to access a resource via Rest / SOAP API

**Authorization: gives role /permission for the resource**

* Almost every REST API must have authorization to check whether you are authorized client or not? before allowing access to the resource
* Authorization is present in header of the request.

**There are 4 types of authorisation:**

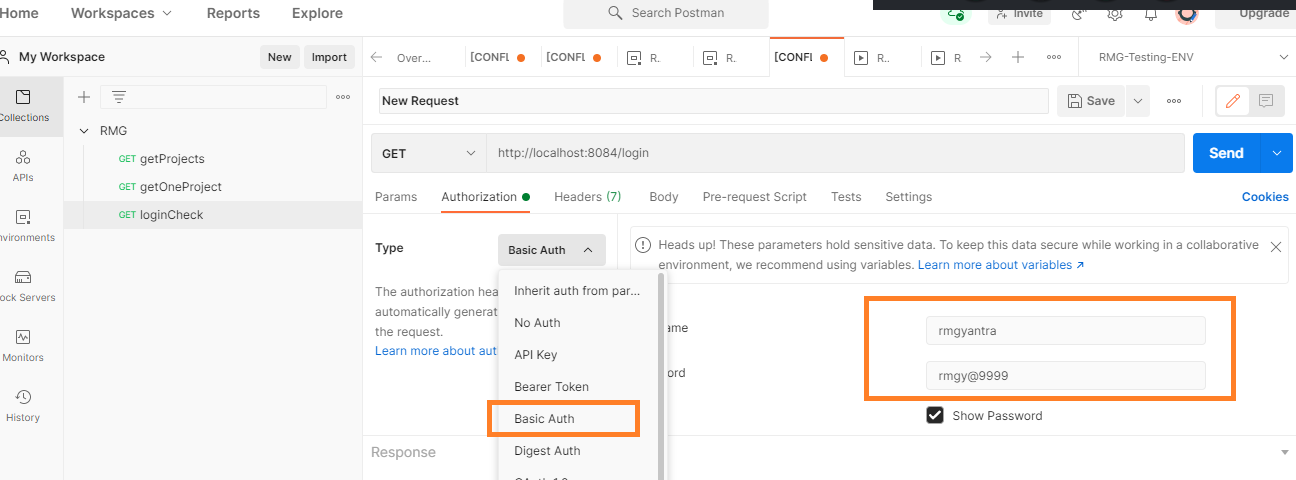
1. **Bearer token**: Required “TokenID” to access an API
2. **Basic auth**: Required “Username & Password “to access an API

**3**. **OAuth 1.0**: Two level authentication

1. **OAuth 2.0**: One level authentication

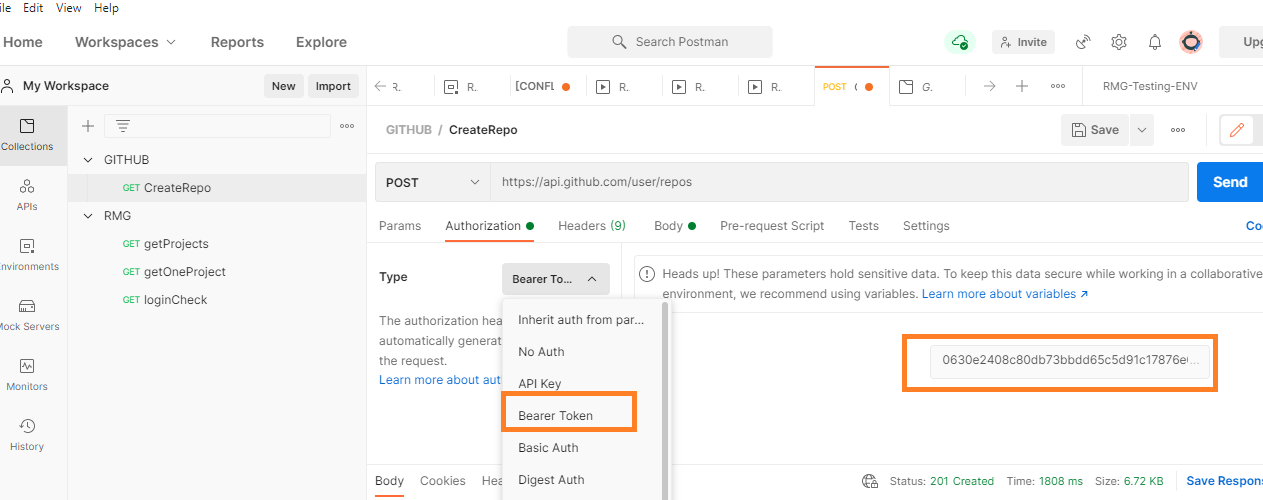
**OAuth**: is an Open Standard for Authorization protocol, widely used in web application, which allows API services to accesses user data without sharing is password.

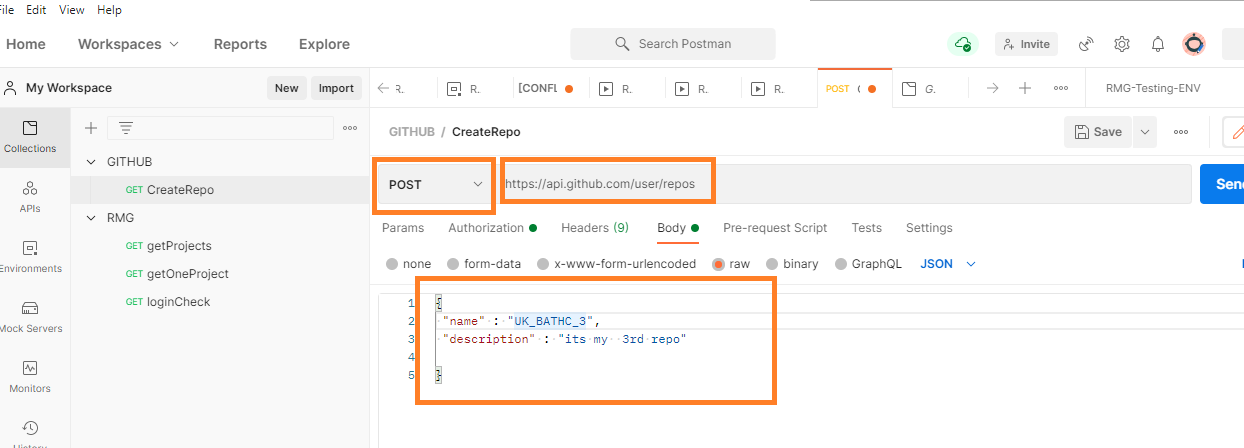
**Basic Auth**: used to access the resource (API) via username / password is called basic auth



Disadvantages: sharing the username /password is not advisable via API, some body can hack your Data

**Bearer Token:** used to access the resource via bearer token ID, that is unique Id to access all the API resource.

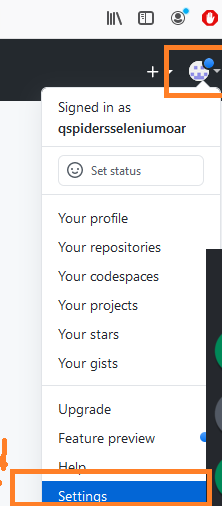




Disadvantages: bearer can also share to other team members, & bearer token id will be fixed until developer changes.

How to create Bearer Token in GITHUB?

1. Go to gitHUB URL <http://github.com>
2. Create an Account with GitHUB
3. Login to Git HUB
4. Go to user Menu 🡺 click Settings



5.click on “Developers settings”

6. Click on “personal Access token”

7. Click on “Generate new Token” button

8. Enter the name & select “Repo” CheckBox

9. Click on “Generate a Token”

10. Copy the Token ID

How to get API for create Repository in GITHUB

1. Go to GIT API Functional Spec Document

https://docs.github.com/en

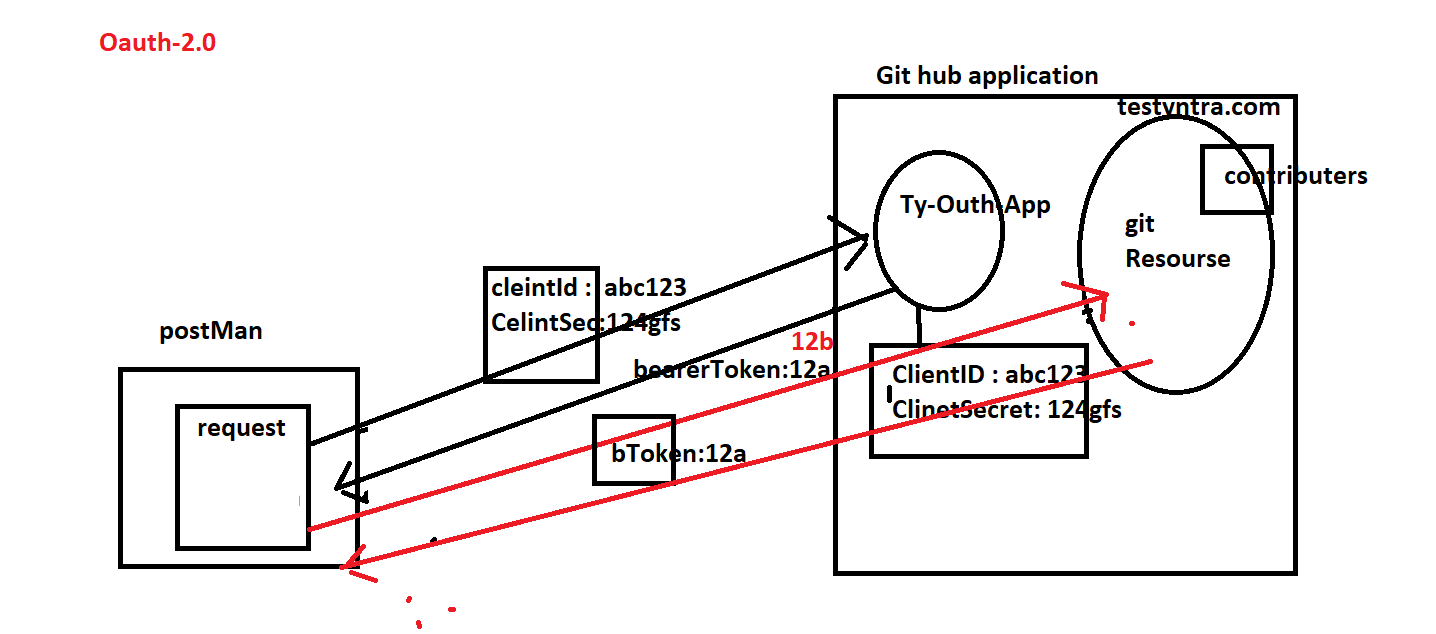
1. Copy the Base URI

https://docs.github.com/en/rest/overview/resources-in-the-rest-api

1. Copy the End PONT & get the JSON BODY structure

https://docs.github.com/en/rest/reference/repos#create-a-repository-for-the-authenticated-user

**OAUTH**-2:

* Whenever your application requests private user data, it must send an OAuth 2.0 token along with the request. Your application first sends a client ID and, client secret to obtain a token. You can generate OAuth 2.0 credentials for web applications, service accounts, or installed applications.
* One level of Authentication
* Client application includes “client secret” with every request.
* 

**OAUTH-1 :**

* **Consumer key and secret & Access Token / Access Secret**: These two strings are used to identify and authenticate the client application
* **Two level of Authetication**

