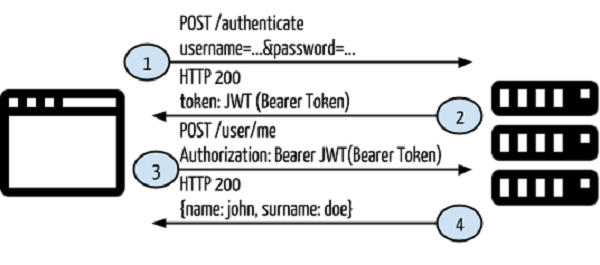
***TOKEN BASED AUTHENTICATION***

***INTRODUCTION:***

The Token-based authentication here **“Cookies and Session”** will not be used. Allow users to enter their username and password in order to obtain a token which allows them to fetch a specific resource - without using their username and password.Tokens are the best way to handle authentication for **“Multiple-users”.**



***WHO USED TOKEN BASED AUTHENTICATION? :***

This application will be used in ***“Web (or) API “***Application like “***FACEBOOK, TWITTER, and GOOGLE PLUS*** etc.….”.

***WHY WE USE TOKEN BASED- AUTHENTICATION? :***

Authentication is the process of determining whether someone or something is actually who or what it declares itself to be. When a potential subscriber accesses an authentication server, a username and ***“PASSWORD*** “may be the only identifying data required.

***BENEFITS OF TOKEN-BASED-AUTHENTICATION:***

* Minimal involvement need from users.
* No additional hardware needed.
* Easy to Manage.

***IMPLEMENTATION OF TOKEN-BASED AUTHENTICATION:***

The App Secret is used in some of the Login flows to generate access tokens and the Secret itself is intended to secure usage of your App to only those that are trusted. The secret can be used to easily create an App Access Token which can make API requests on behalf of any user of the app, which makes it extremely important that an App Secret is not compromised.

Therefore the App Secret or an App Access token should never be included in any code that could be accessed by anyone other than a developer of the app. This applies to all methods of code that are not secured like client-side code (such as HTML or Javascript) or native apps (such as iOS, Android or Windows desktop apps) that could be decompiled.

We recommend that App Access Tokens should only be used directly from your app's servers in order to provide the best security. For native apps, we suggest that the app communicates with your own server and the server then makes the API requests to Facebook using the App Access Token. For this reason, if you’re 'App Type' under Advanced Settings in the App Dashboard is set to Native/Desktop we assume that your native app contains the App Secret or an App Access Token in the binary, and we do not allow calls signed with an App Access Token to proceed. The API will behave as though no access token was provided.

If your App Secret is compromised, you should reset it immediately in the Basic Settings of your App Dashboard. When you start the reset process, you can specify a number of hours that the compromised secret will continue to work for when making requests, however anything sent from Facebook (such as signed requests) will use the new secret straight away, so you must adjust your code to expect it as soon as possible.

SOLUTION:

A generic token is a random string. Old token can be removed automatically in order to prevent the server’s database from growing databases from growing indefinitely. The token must be allowed to be used only once. It must only be usable for the user it was created for.It must only be sent Via HTTPs,.It should expired date on 7 days.

***BENEFITS:***

* It can be Stateless and Scalable.
* Extensibility.
* Multiple Platforms and Domain
* Standards Based.

