OAuth 2.0 is an open-standard authorization framework that allows services or servers to provide delegated and regulated access to their assets.

OAuth 2.0 - (open authorization (getting limited access to resources))

This is where you give another entity, say an app on your cell phone, access to your resources. For example google Map wants to have access to your location, and it asks your permission. and after you granted, it will use only your location(and not the others like photos, calls etc.). So theris authentication (you have to unlock your phone) and authorization(you gave access to use location to google Map app).

One more example: theris a web page “photosW '' that is a container for photos, again there is folder with public photos the one that you use for public social networks(instagram, VK, so on) and private folder for you only (may be there are photos that are private for you and so far you don't want to share it). For you to access your photos (both public and private) you use a username and password. Here are resources and you are the resource owner. Now let's say that you want to give access to somebody else only to your public folder. So you have somebody else that you may call a client or you may consider it as an app, so what you are doing is giving this client authorization. Authorization meaning limited access to your recorcess. This client or APP is authorized to use your photos, not to create any photos in there, not to delete, but only to get your public photos. It cant access your private photos because you only authorized to see your public photos. It also cant get your username and password, because if it happens, this client will have access to everything: all photos, ability to create/delete, do anything that it wants.

So the idea of Oauth, meaning other authorization (OAuth). You are giving to other clients the authorization that gives limited access to some resources. OR: 1)-allows 3-rd party access. 2)-When it says limited access it means-authorized. 3)-It gates limited access to web service, to API.

All of OAuth 2.0 is HTTP request only you can't use anything else

Since there is a problem with giving your credentials (username+password) to 3-rd party, the third party is given an access token. The 3-rd party uses this access token to get authorized access (that is limited access) to your resources, a HTTP must be used when going through all.

**Access token** - is used to get the protected resources. The whole point of the OAuth 2.0 is to get an access token to the client so that the client can use it to get authorized access to the resource owner`s resources. It has a String format.

Roles/terminology in oAuth:

- **Resource owner** (person ho owns the content)

-**Resource server**, thats what provides photos when given access token (I think its API).

-**Client / Application**, the 3-rd party that wants access to what you own as the resource owner.

-**Authorization server**, is what provides access token to the client or APP to get the resources. It is typically coupled (соединенный) with a resource server, It can be a separate server or it can be the same server.

**SERVER**-we have a computer out there called Windows server, but this window server is not a server - this is a computer on which there are servers (list of services). And these services are either running or not. And why it is important for this servers / services to run? When its running its looking for something to come in so it can serve. That's why its call server because it waiting for the client to serve. Say, for example client trying to get access to public photo. the API has a running server waiting for request because its running, its always looking for this request. If it's stopped its might be put another server redundant (типа избыточный/добавочный).

Authorization Grant Flow:

PROTOKOL FLOW FOR OAuth 2.0

1)- Client requests limited resources from the resource owner. It sends it in the form of an authorization request.

2)- The resource owner says yes you have a grant, we enable you to access the resources

3)-The client than sends an authorization grant to the authorization server that is responsible for resources.

4)-The authorization server than sends back to client an access token.

5)-Than client sends access token to resource server.

6)-The resource server then sends to the client requested content.

There 4 kinds of authorization grant: (то что описано во 2-ом пункте выше).

1)-authorization code flow is where the client asks the resource owner for permission through the authorization server.In that case the resource owner has to to log into the authorization server if not already logged into and than say ok. The client does not ask the resource owner directly. This case is the most used.

2)-Implicit is were a call is made for the resources by Ajax or JavaScript.That is code in HTML itself.

3)-Resource owner is where resource owner or somebody who has full permission.

4)-The client credentials method is where the resource owner is never asked for permission

Flow of authorization code grant (detailed description of getting code grant)

Before this flow starts the client must register with the authorization server and get a client ID.

The user clicks on authorization endpoint and authorization endpoint points to to the authorization server. When the user on the browser goes on the authorization server it brings alone with it client ID and the redirect URI (is where to sent the user back to the client after the user authenticates. So the user and the client starts at the client. The client specifies the link to the authorization endpoint that is the authorization server ← вот эта часть не понятна

ok one more time about OAuth - it is when you dont access the API but you give responsibility to another app to run the API. The same as would give permission to the navigator to use your location on the phone.

The difference between basic authorization and OAuth. When use basic authorization you send password and username to the server. When it is OAuth there is an app that has your permission (authorization - limited access) to access the API using your password and username and give you the result, you don't need to do anything.