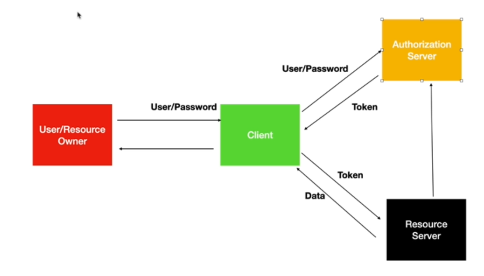
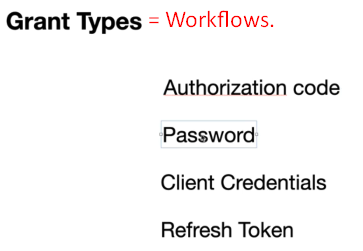
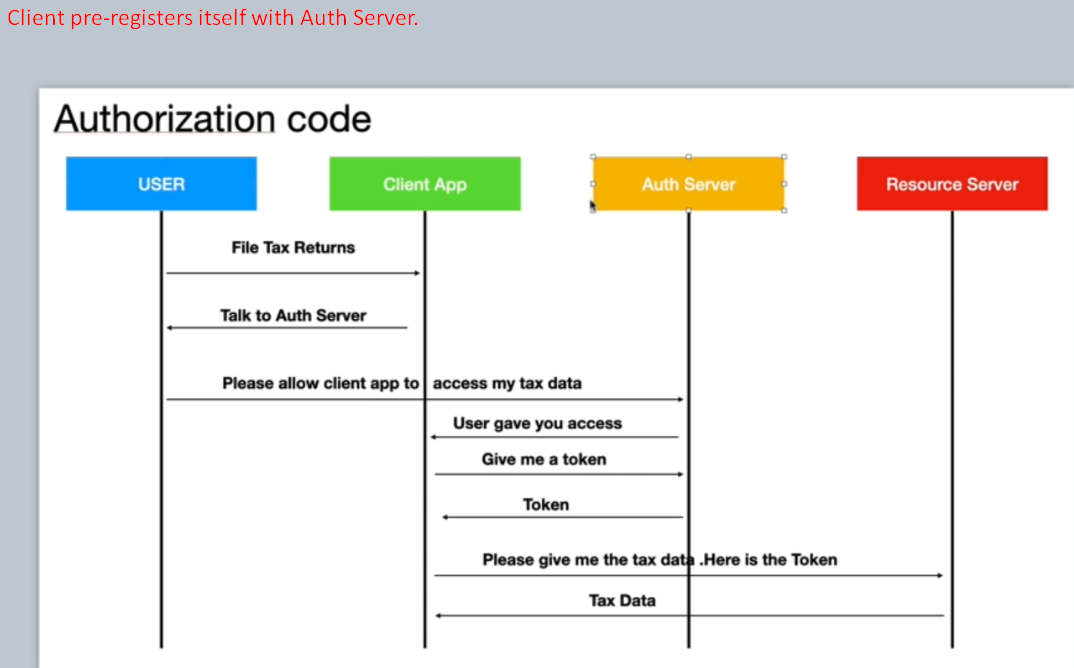
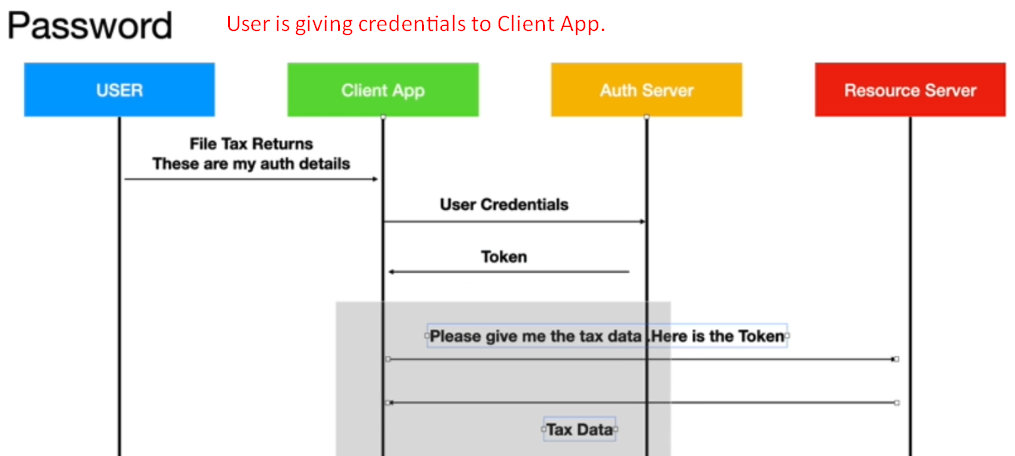
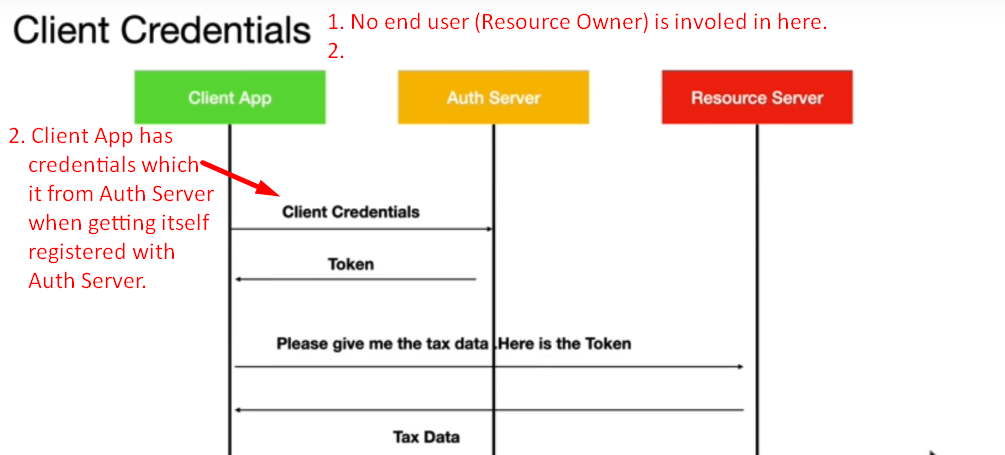
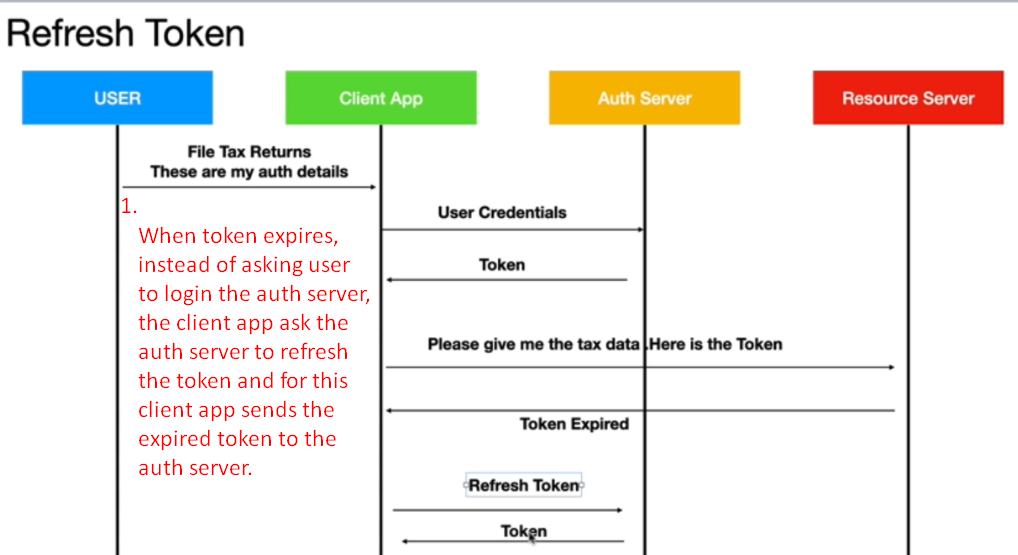
1. 
2. Oauth supports multiple workflows depending on how user id and password are exchanged b/w user and the client application or how the communication b/w user and authorization server happens.
3. 
4. **Authorization Code Grant Type**: 
   1. This is where a user (resource owner) will grant a client on his behalf but will not do it directly.
   2. **Steps**: Let’s understand the sequence diagram.
      1. **File Tax Returns**: A user accesses a web page on client’s web app to fill his return.
      2. **Talk to Auth Server**: The client app redirects the user to Google auth server as it needs some docs.
      3. **Please allow client app to access my tax data**: The user enters his credentials on the redirected page from Google.  
         **NOTE**: When redirecting to a Google Page, the client also sends his client id and secrete (password).  
         **NOTE**: These client id and secrete are given to client when a client gets itself registered with Google auth server.
      4. **User gave you access**: At this point of time, authorization server gives only authorization code but not an access token.
      5. **Give me a token**: At this point of time, the client will give authorization code, client id and secrete (password) to the authorization server and the authorization server will give back an access token.
      6. **Please give me the text data**. **Here is the token**: Then client sends the client id, secrete and access token to resource server to get the resource and then resource server verifies the token with authorization server and on successful, returns the resource.
5. **Password Grant Type**:  
   
   1. The only difference b/w the Authorization Code Grant Flow and Password Grant Flow:
      1. Authorization Code Grant Flow:
         1. User is redirected to Authorization Server’s login page where he enters the credentials. So not giving credentials to client.
      2. Password Grant Flow:
         1. User gives credentials to client itself and client uses the credentials to get a token.
6. **Client Credentials Grant Type**: 
   1. NOTE: Here we don’t have any user in picture. Typically, you can think of it as an application accessing another app. Very typical in micro services.
7. **Refresh Token**:
8. 
   1. So far in all kinds of grant types, the access token has a life time (expiration time) for security reasons.
9. We can create our own grant type too.