OAuth –

“Outside” authentication?

Explore using third-party web services using OAuth

1. Authorization requires authentication
2. OAuth
   1. A communication protocol for third-party access to other servers
   2. Logging in via the authorization logic of third parties
      1. Google, FB, and Github invest heavily into their authorization processes
      2. Therefore these authorizations are likely far more secure than ones that a smaller dev team can come up with.
      3. Potentially access data, depending on permissions
   3. We’ll be going over the workflow for Github OAuth.
   4. This is going to be a communication protocol.
   5. The set-up process will be different depending on what API we decide to use.
   6. We’ll have to get real real comfy with docs for implement our OAuth as we need to.
3. Before we can build authentication, we need to register the app with the OAuth provider
   1. Client\_id
   2. Client\_secret
   3. redirectURI (callback URL)
4. Go to github
   1. Go down to settings
   2. Go down to developer settings
      1. Click “OAuth” app on left bar
      2. Register your app
         1. Give your application a name
         2. The URL for the page.
            1. We can use Localhost:3000 for this part
         3. A Description
         4. An authorization callback URL
            1. This is the URI….
            2. When a client is uses github to login, they are redirected to github
            3. And once signed in, they need to know where to send the client. This is what the CB URL is for.
            4. WE WILL NEED TO SET UP A ROUTE HANDLER FOR THIS PART.
         5. Client ID will be provided after we register our application.
            1. We will be placing this client ID somewhere on the request we make
         6. Client secret is generated once we create OAuth app.
            1. WE NEED TO SAVE THIS SOMEWHERE.
            2. WE DO NOT PLACE THIS IN SERVER OR CODE
            3. WE PLACE THIS SOMEWHERE LOCAL AND SAFE.
   3. There are 3 players in our OAuth procedure.

|  |
| --- |
| GITHUB |

|  |
| --- |
| Client |

|  |
| --- |
| Server |

5B – send back user data

5A – requests for user data

2a – redirect to github

Client\_id & redirect\_URI

3A – Redirect client to server with

Auth\_code

1. Login with Oauth

2a – redirect to github; has client\_id

3B– Redirect client to server. Check auth\_code

4B – Github sends token to server to be stored as authentication for future requests to Github

4A– send auth\_code, client\_id, client\_secret to be proofed by github

1. The token returned from the OAuth provider is what grants us access to the scope of the API we requested.
   1. Future requests contain the API URL and a header with the access token.
   2. It does not have to be implemented in this way, but this token can be used as a session token.
   3. The URI is the same URI we pass.
   4. Github will not be looking at our account to check the URI – so we need to pass the URI.
      1. The client\_secret; **we are the client for github at this moment.**
      2. The client\_id: **identifies our application to github.**
2. Pros and Cons