

DeSo Developer Hub

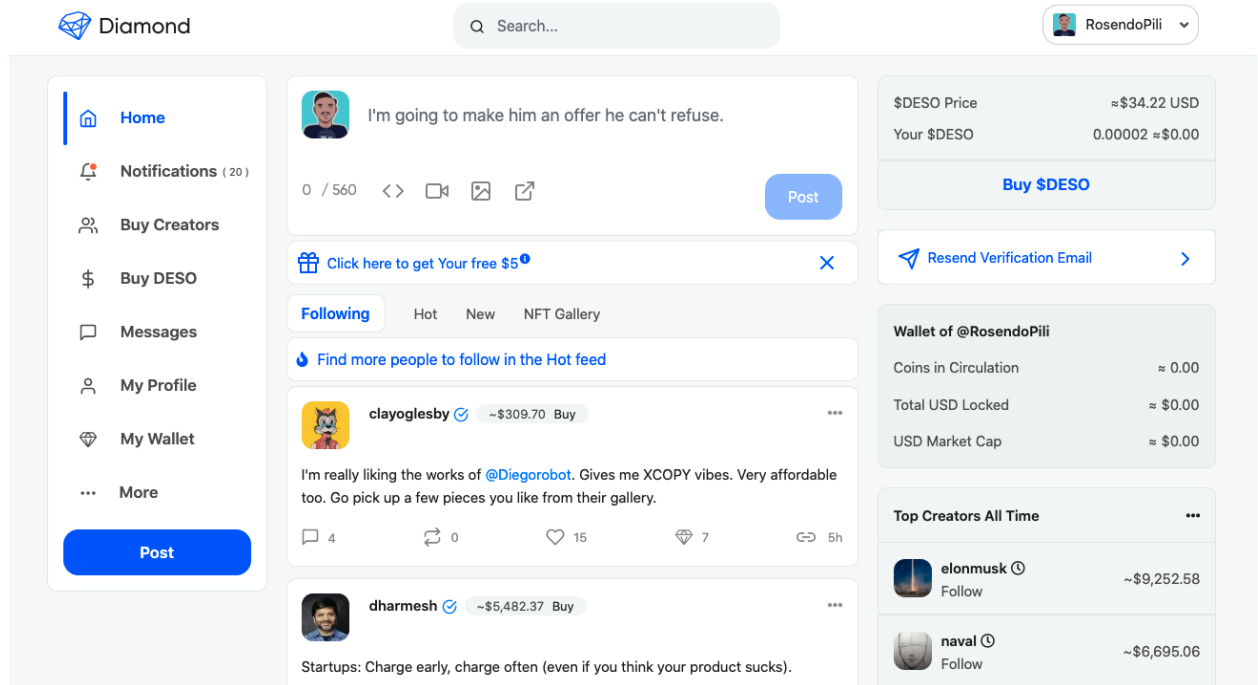
Getting Started Guide

Howdy y'all,

The DeSo team has recently released their developer hub, designed to make all of DeSo's API functionalities more accessible to hackers like you.

Below you will find a step by step guide on how to get started with using the [DeSo Developer Hub](#).

1. Head over to the diamond app at diamondapp.com to create an account and receive free \$DESO coin. You will need to do this if you want to write to the DeSo Blockchain.
2. Once you create your account you should see a page similar to the following screenshot:



3. Now, let's open up the a terminal and call

```
git clone https://github.com/DeSoDog/deso-react.git
```

to install a preconfigured deso-react starter app. (If you do not want to use react you will need to create your own web application and then call "npm i deso-protocol" to add the DeSo Library)

- Once the repo is downloaded run the following commands

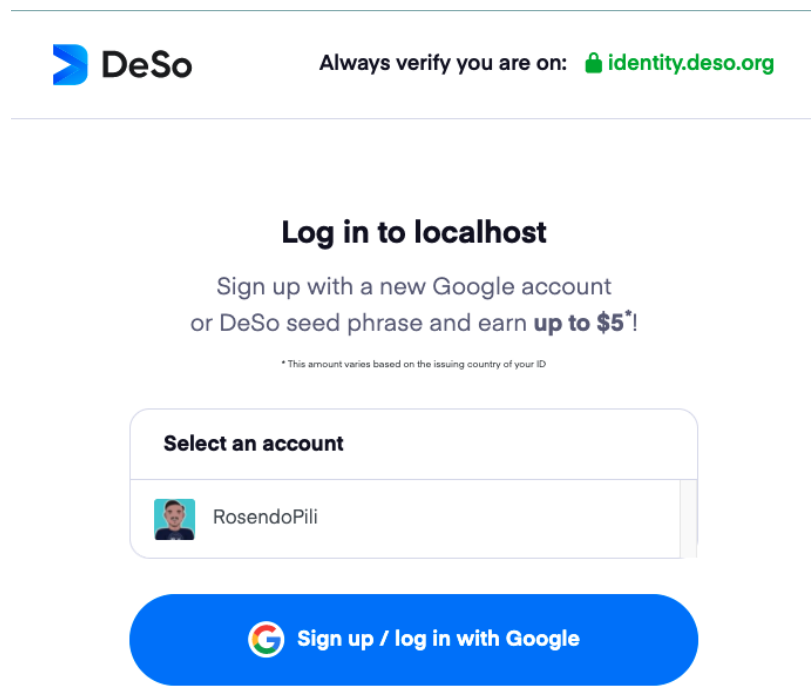
```
cd deso-react  
npm i
```

to install your dependencies.

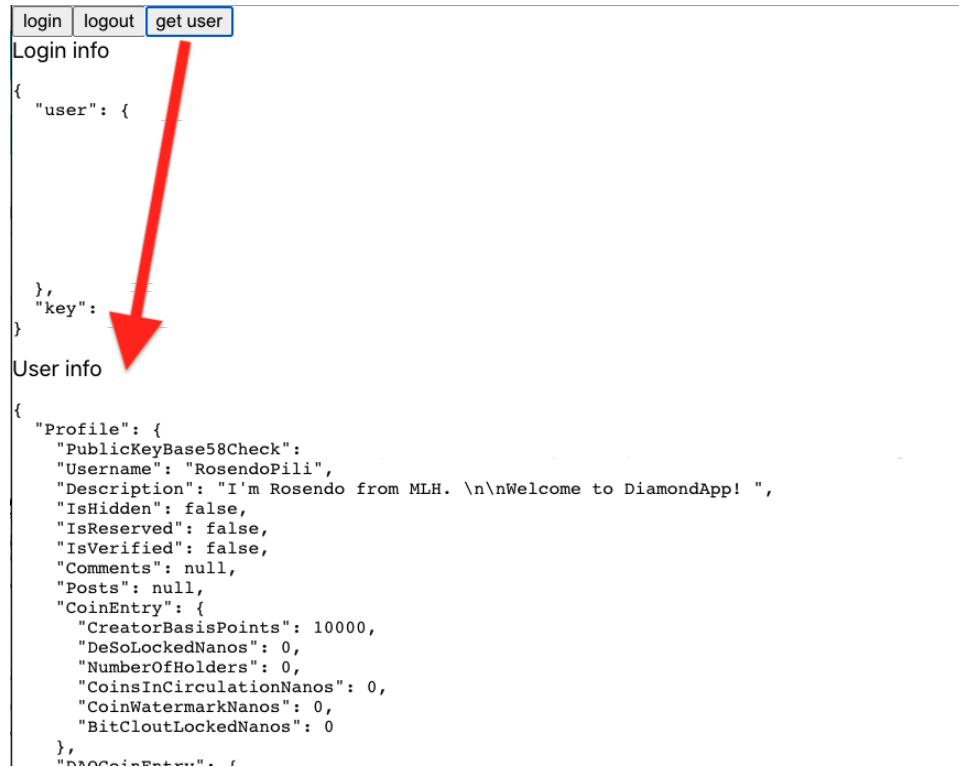
- To run the application call `npm run start`
- Go to `localhost:3000` in your browser and you should see 3 buttons that demonstrate the login functionality, logout functionality and a get call to retrieve user information.



- Since we created a DeSo account in the Diamond App, clicking on the login button will open a login modal and display the account you recently created. Click your account and the application will now have a reference to it.



8. If you click the 'get user' button, the library will make a call to the blockchain and display information about your account (your screen will display your user information).



9. To see how these calls work head over to app.js

TLDR; these buttons are bound to some onClick functions that call to the DeSo API. If you want to see the contents of the deso object instantiated on line 4, run a

console.log(deso) on line 5 and view the object in your browser console.

```
deso-react > src > JS App.js > App
1 1 import './App.css';
1 2 import Deso from 'deso-protocol';
0 3 import { useState } from 'react';
9 4 const deso = new Deso();
8 5
7 6 function App() {
6 7   const [sampleResponse, setSampleResponse] = useState();
5 8   const [loginResponse, setLoginResponse] = useState();
4 9   return (
3 10     <div>
2 11       <button
1 12         onClick={async () => {
0 13           const user = await deso.identity.login();
1 14           console.log(user);
8 15           setLoginResponse(JSON.stringify(user, null, 2));
7 16         }}
6 17       >
5 18       login
4 19     </button>
3 20     <button
2 21       onClick={() => {
1 22         deso.identity.logout(deso.identity.getUserKey());
0 23       }}
9 24     >
8 25     logout
7 26   </button>
6 27   <button
5 28     onClick={async () => {
4 29       const user = await deso.user.getSingleProfile({
3 30         PublicKeyBase58Check: deso.identity.getUserKey(),
2 31       });
1 32       console.log(user);
0 33       setSampleResponse(JSON.stringify(user, null, 2));
34     }}
35   >
36   get user
37 </button>
38
39   <div>
40     Login info
41     <pre>{loginResponse}</pre>
42   </div>
43   <div>
44     User info
45     <pre>{sampleResponse}</pre>
46   </div>
47 </div>
48   );
49 }
50
51 export default App;
52
```

10. Next, let's head back to <https://hackp.ac/deso-devhub> and use their sample code to implement some new functionality to our react-app. Simply find the endpoint that you want to implement, look at the example and get building. Here's an example of the submit post page, which gives all the info you need to know to make a submit post request.

LANDING	OVERVIEW SUBMIT POST
Welcome	Submit Post Submit a post
IDENTITY	Using Submit Post See additional parameters
Login	<pre> 1 import Deso from 'deso-protocol'; 2 const deso = new Deso(); 3 const request = { 4 "UpdaterPublicKeyBase58Check": [REDACTED], 5 "BodyObj": { 6 "Body": "Checking out the developer hub", 7 "VideoURLs": [], 8 "ImageURLs": [] 9 } 10 }; 11 const response = await deso.posts.submitPost(request); </pre>
Logout	
JWT	
NFT	
Get Nfts For User	
Get Nft Bids For User	Try It Out Click here to call Submit Post.
Get Nft Bids For Nft Post	
Get Nft Showcase	
Get Next Nft Showcase	

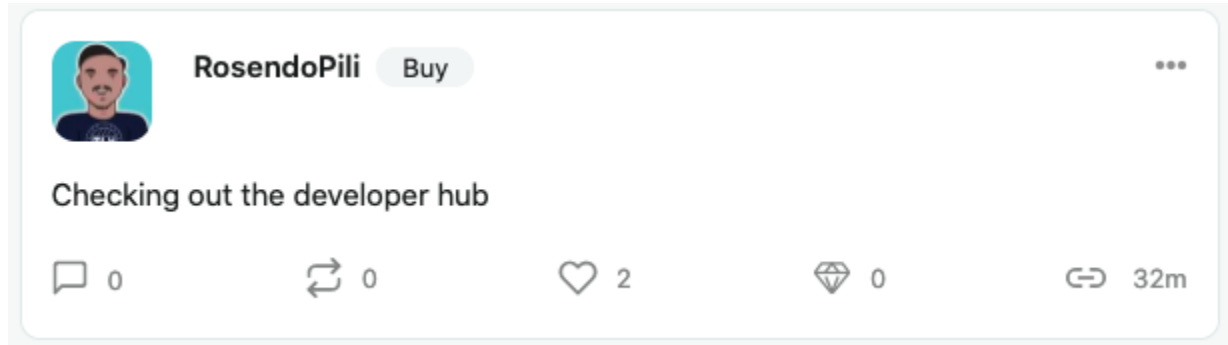
11. Using the 'submit post' code we see displayed but replacing the public key with a reference to the deso API object, we're going to build a new button with the onClick functionality that will allow us to submit a post using our react-app. Beneath the 'get user' button, copy and paste the following code:

```

<button onClick={async() => {
  const request = {
    "UpdaterPublicKeyBase58Check": deso.identity.getUserKey(),
    "BodyObj": {
      "Body": "Checking out the developer hub",
      "VideoURLs": [],
      "ImageURLs": []
    }
  };
  const response = await deso.posts.submitPost(request);
}
}> submit post
</button>

```

12. Once your code app re-renders, you should be able to see a 'submit post' button to the right of your 'get user' button. When you click it, it will submit a post on your DiamondApp account that should look like the following screenshot. Remember, you can always change the body of the request to say what it is you want to post.



And there you have it. You're officially a blockchain developer, but don't go updating your linkedin profile just yet. There are still a ton of other features you can try out on your own and use in your next project!

That's it for now.

The purpose of the DeSo Developer Hub is to give hackers the opportunity to try out the DeSo API endpoints in a low-stakes environment. The accompanying starter react-app gives them the ability to utilize the API in a more customizable way and eventually build their own applications.

If you have any issues with the documentation, you can add them at hackp.ac/deso-feedback