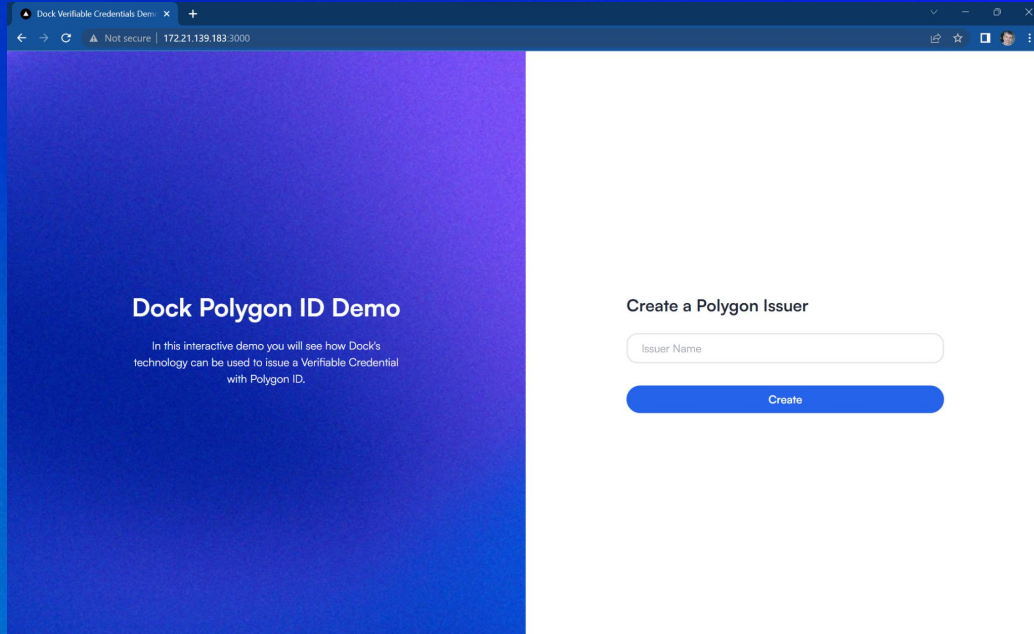


Issuing Polygon ID Credentials with Dock

July 21, 2023



Dock Polygon ID Demo App



<https://github.com/docknetwork/polygonid-demo>

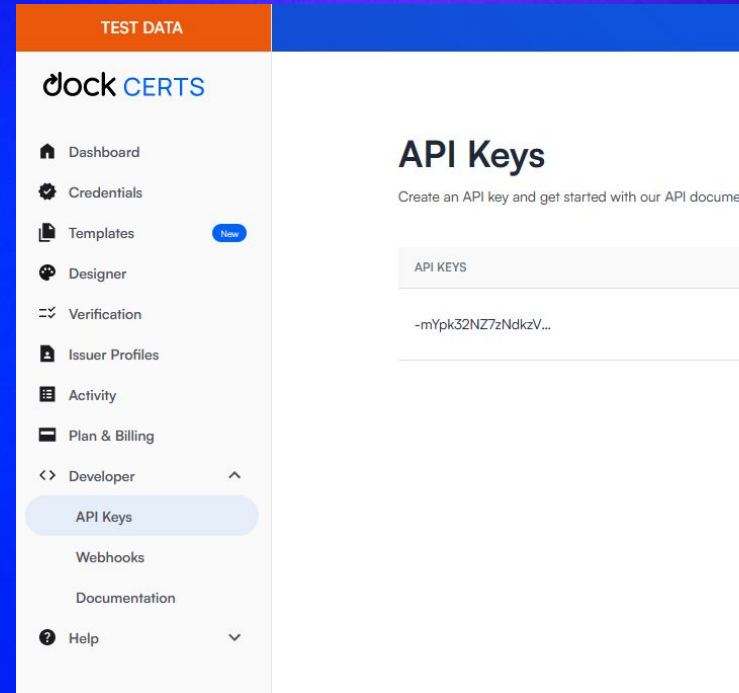
Issue VC's in Four Steps

1. Get an API key from Dock Certs
2. Create a Polygon ID issuer profile
3. Create a credential claim request
4. Share the claim QR code link with the recipient

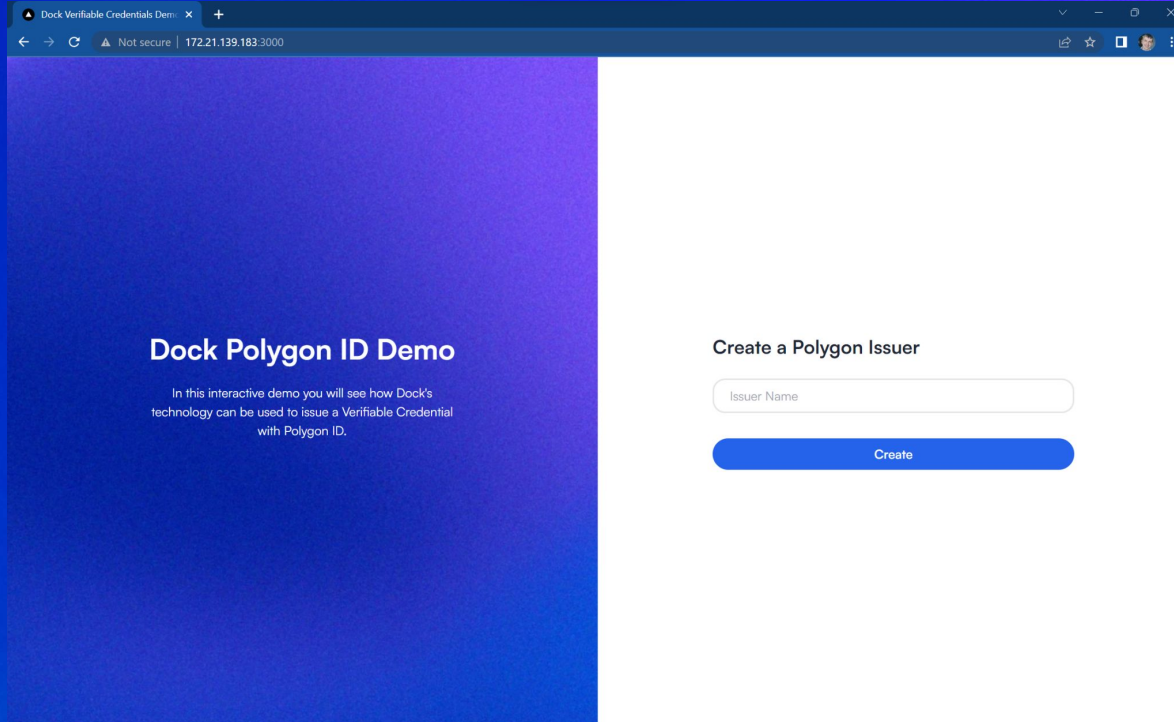


1. Get an API key from Dock Certs

1. Sign up for a free trial at <https://certs.dock.io>
2. Expand the `Developer` section on the left navigation panel
3. Click the `Create API key` button
4. Copy the API key to your project
 - a. E.g. Update the `DOCK_API_TOKEN` value in your `.env` file



2. Create a Polygon ID issuer profile



The screenshot shows a web browser window with the title 'Dock Verifiable Credentials Demo'. The address bar indicates a 'Not secure' connection to '172.21.139.183:3000'. The page is split into two main sections. The left section, with a purple-to-blue gradient background, is titled 'Dock Polygon ID Demo' and contains the text: 'In this interactive demo you will see how Dock's technology can be used to issue a Verifiable Credential with Polygon ID.' The right section, with a white background, is titled 'Create a Polygon Issuer' and features a single text input field labeled 'Issuer Name' and a blue 'Create' button below it.

Dock Polygon ID Demo

In this interactive demo you will see how Dock's technology can be used to issue a Verifiable Credential with Polygon ID.

Create a Polygon Issuer

Issuer Name

Create



2. Create a Polygon ID issuer profile

Create a Polygon ID DID

1. Set DOCK-API-TOKEN header
2. Set keyType to 'bjj' for Baby Jub-Jub
3. Set type to 'polygonid'
4. POST the request to <https://api-testnet.dock.io/dids>

```
const axiosHeaders = {  
  headers: {  
    'DOCK-API-TOKEN':  
process.env.DOCK_API_TOKEN,  
  },  
};  
  
const polygonDidBody = {  
  keyType: 'bjj',  
  type: 'polygonid'  
};  
  
const didResp = await  
axios.post(`${baseUrl}/dids`,  
polygonDidBody, axiosHeaders);
```



2. Create a Polygon ID issuer profile

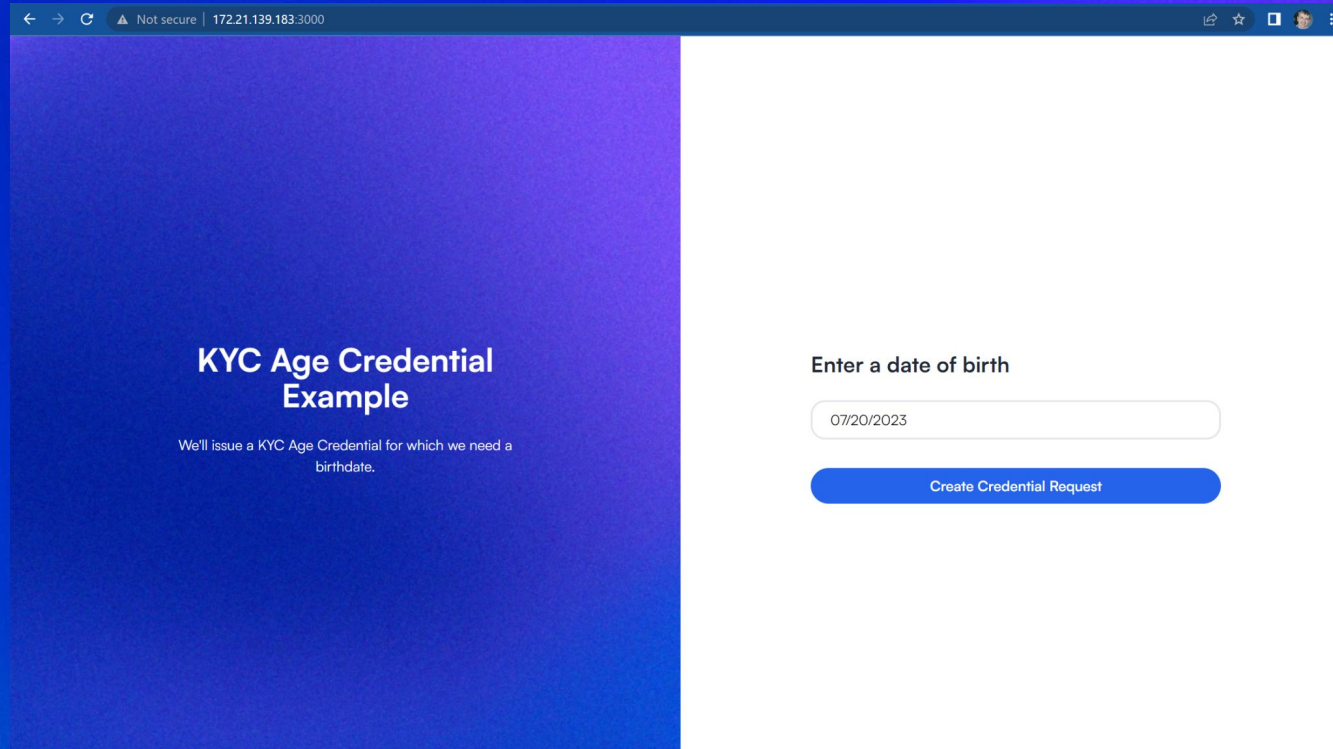
Create an Issuer Profile (optional)

1. Set DOCK-API-TOKEN header
2. Set the `name` and `desc` field to meaningful values
3. Set `did` to the DID created in the previous step
4. POST the request to <https://api-testnet.dock.io/profiles>

```
const reqBody = {  
  "name": "My Test Polygon ID DID",  
  "did":  
    "did:polygonid:polygon:mumbai:2qFuT3DDK6M7k8  
    hYozPZMd8ouN8nY2sJ8Fivriw5YQ",  
  "description": "Testing out the Issuer  
Profiles api"  
};  
  
const profileResp = await  
axios.post(`${baseUrl}/profiles`, reqBody,  
  axiosHeaders);
```



3. Create a credential claim request



The screenshot shows a web browser window with a dark blue header bar. The address bar displays "Not secure | 172.21.139.183:3000". The page is split into two main sections. The left section has a dark blue background with white text that reads "KYC Age Credential Example" and "We'll issue a KYC Age Credential for which we need a birthdate." The right section has a white background and contains a form. The form has a label "Enter a date of birth" above a text input field. The input field contains the date "07/20/2023". Below the input field is a blue button with the text "Create Credential Request".

← → ↻ ⚠ Not secure | 172.21.139.183:3000

KYC Age Credential Example

We'll issue a KYC Age Credential for which we need a birthdate.

Enter a date of birth

07/20/2023

Create Credential Request



3. Create a credential claim request

1. Set DOCK-API-TOKEN header
2. Set the `schema`
3. Set `claims` to `id`
4. Set `issuer` to the DID created in the first step
5. Set `name` to the name of the schema
6. Include schema name in the credential type array
7. Populate `subject` with credential specific details
8. POST the request to <https://api-testnet.dock.io/credentials/request-claims>

```
const requestBody = {
  schema:
'https://raw.githubusercontent.com/iden3/claim-schema-vocab/main/
schemas/json/KYCAgeCredential-v2.json',
  claims: [ 'id' ],
  credentialOptions: {
    credential: {
      schema:
'https://raw.githubusercontent.com/iden3/claim-schema-vocab/main/
schemas/json/KYCAgeCredential-v2.json',
      issuer:
'did:polygonid:polygon:mumbai:2qFuT3DDK6M7k8hYozPZMd8ouN8nY2sJ8Fi
vriw5YQ',
      name: 'KYCAgeCredential',
      type: [ "VerifiableCredential", "KYCAgeCredential" ],
      subject: {
        birthday: 20230719,
        documentType: 3324
      }
    },
    distribute: true
  }
}
```



Schemas

****** Hard-coded to KYCAgeCredential in this example

You can find other sample schemas on the Iden3 Github repo (<https://github.com/iden3/claim-schema-vocab/tree/main/schemas/json>).

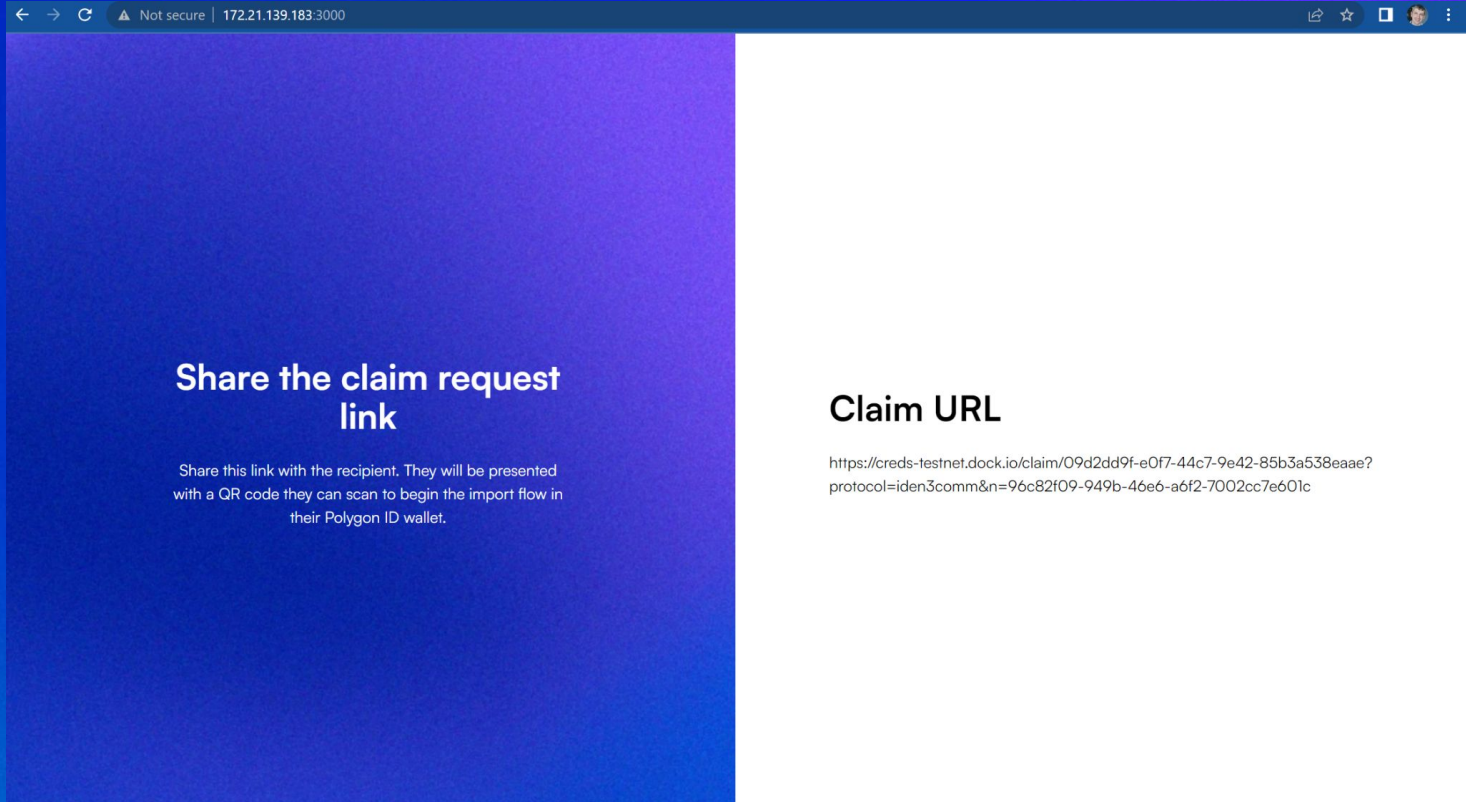
NOTE: make sure you grab the `raw` version of the file.

Custom schema support

- link to a publicly hosted schema directly
- create a custom schema using
 - the Templates (<https://certs.dock.io/schemas>) feature on Dock Certs
 - use the Credential Schemas (<https://docs.api.dock.io/#credential-schemas>) routes in the API



4. Share the claim QR code link with the recipient



The screenshot shows a web browser window with a dark blue header bar. The address bar displays "Not secure | 172.21.139.183:3000". The main content area is split into two panels. The left panel has a dark blue background with white text. The right panel has a white background with black text.

Share the claim request link

Share this link with the recipient. They will be presented with a QR code they can scan to begin the import flow in their Polygon ID wallet.

Claim URL

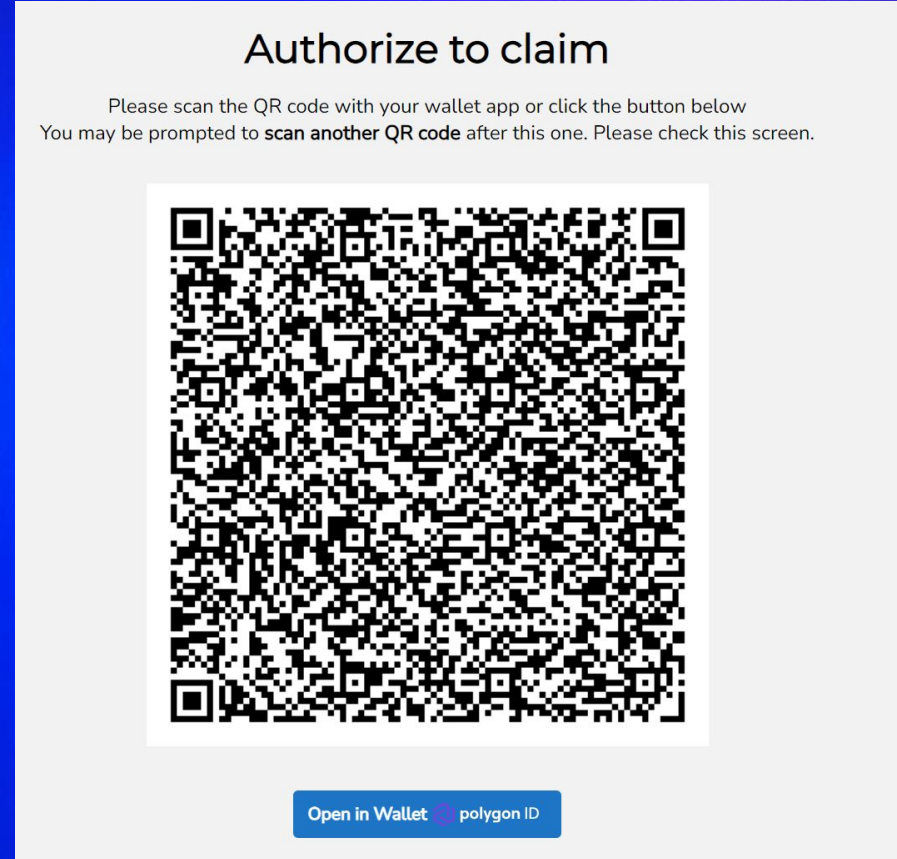
<https://creds-testnet.dock.io/claim/09d2dd9f-e0f7-44c7-9e42-85b3a538eaae?protocol=idn3comm&n=96c82f09-949b-46e6-a6f2-7002cc7e601c>



4. Share the claim QR code link with the recipient

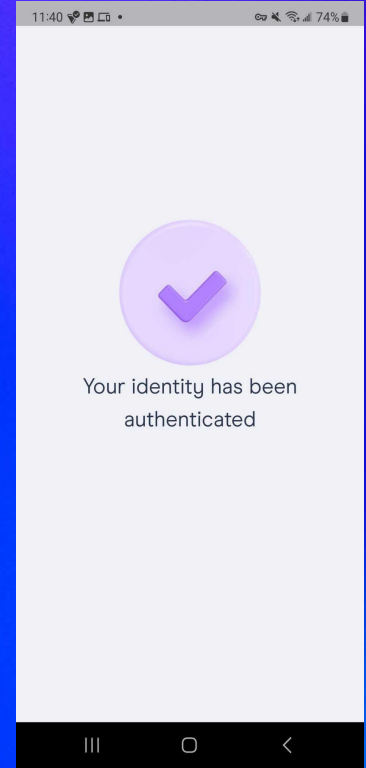
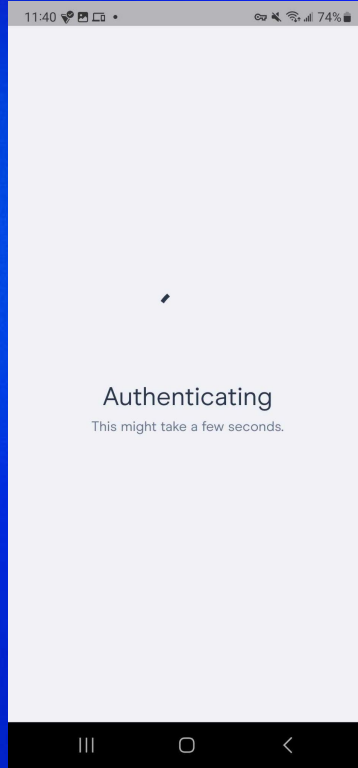
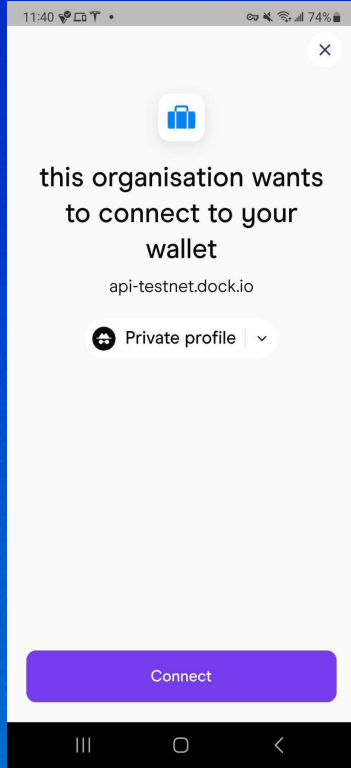
Loading the link found in the 'qrUrl' field in a browser will launch Dock's credential claim flow for Polygon ID

Note: Polygon ID's flow requires two QR code scans



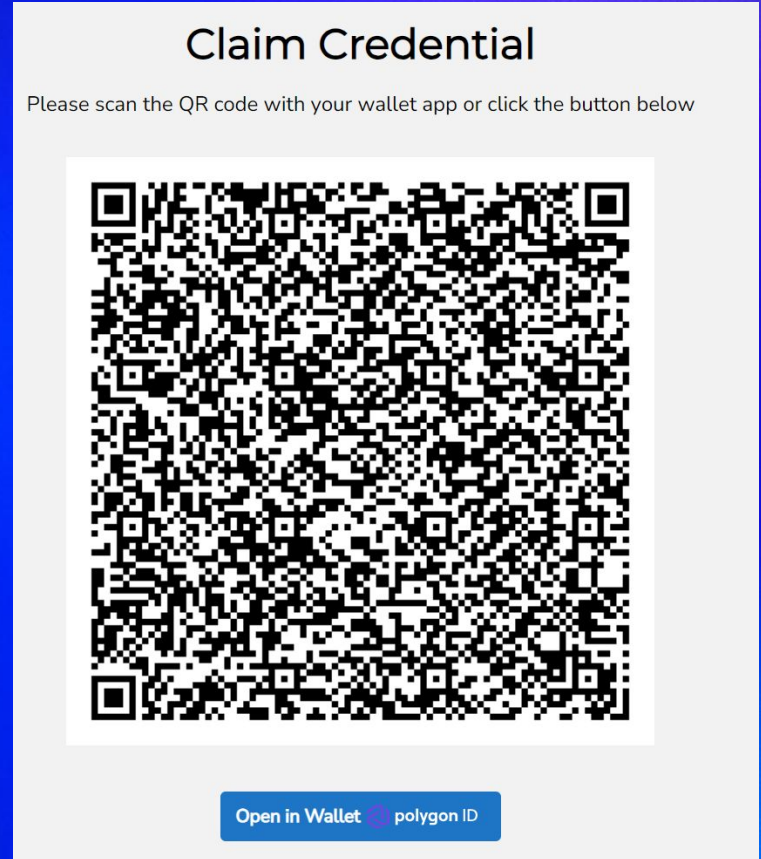
4. Share the claim QR code link with the recipient

After scanning the Authorize to Claim QR code the wallet will authenticate...



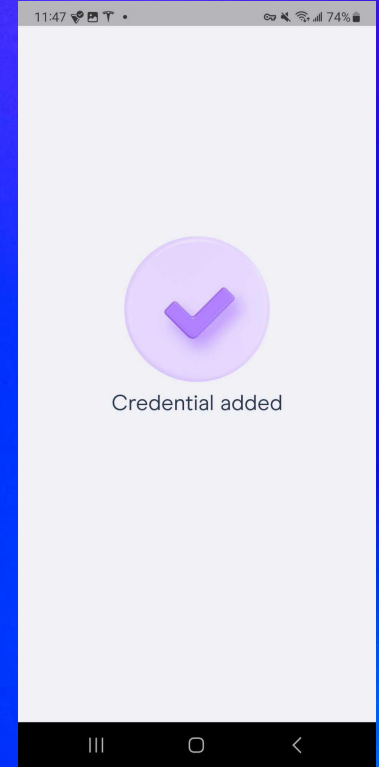
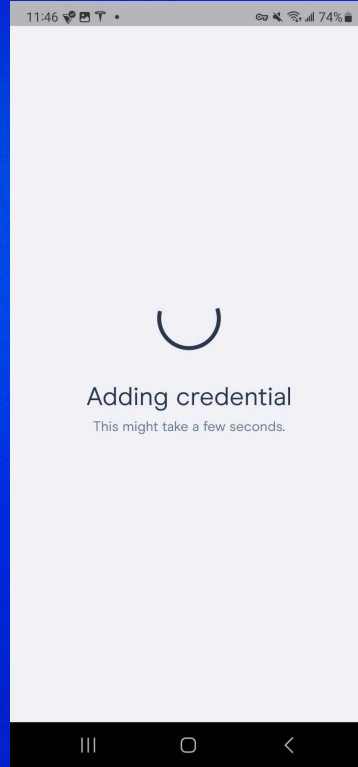
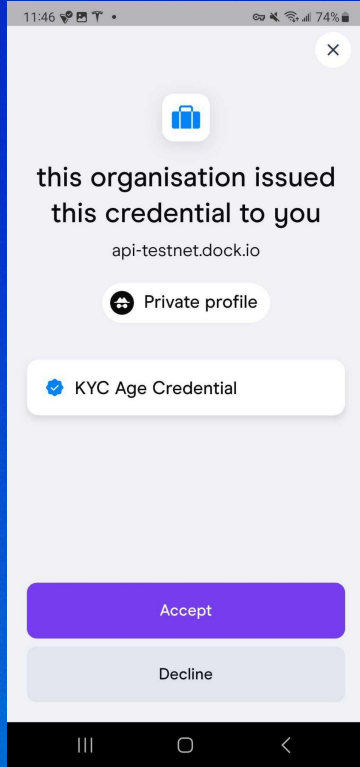
4. Share the claim QR code link with the recipient

After the wallet authenticates the QR in the browser will automatically update.



4. Share the claim QR code link with the recipient

Scanning the Claim Credential QR code will import the credential into the wallet



Dock Polygon ID Demo App



<https://github.com/docknetwork/polygonid-demo>

Dock Certs



<https://certs.dock.io>



Dock API Docs



<https://docs.api.dock.io>

