Track donations with Blockchain

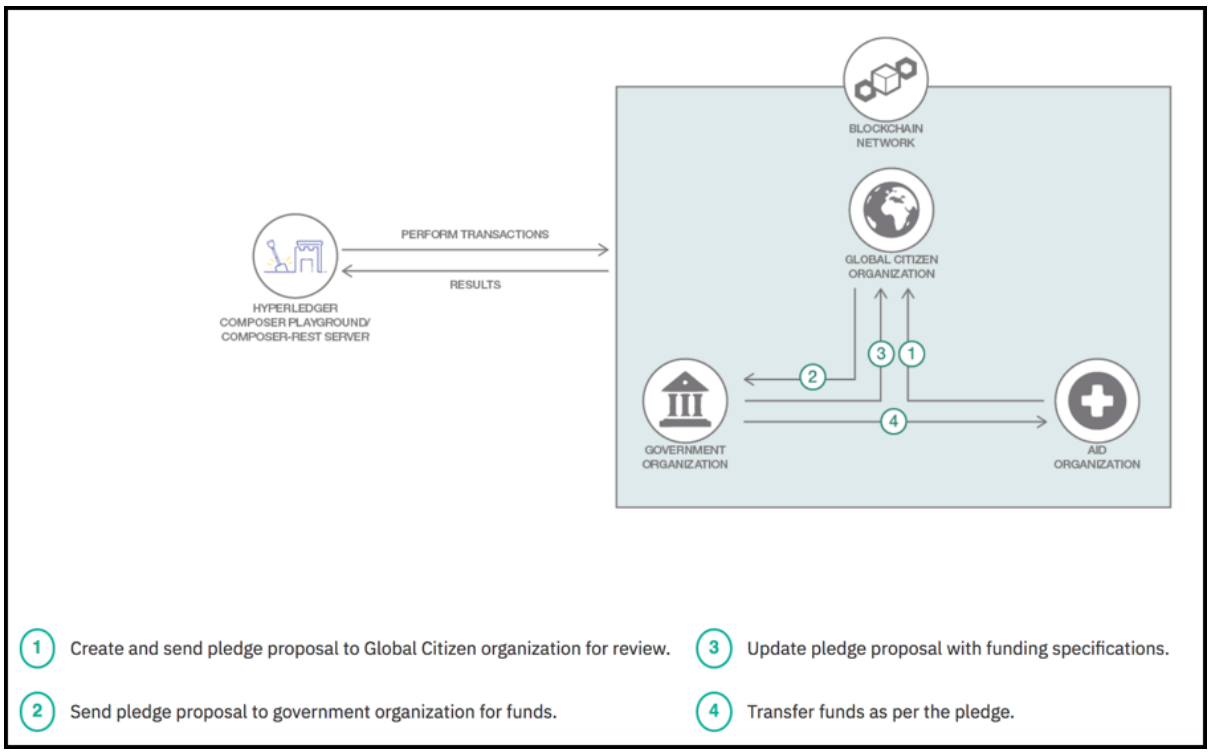
You can find the code at:
https://github.com/AreejEssa/global-citizen
(in the "BC-Workshop" folder – first folder in the repo)

This document is compiled by Hashim Noor (hashim.noor1@ibm.com)

Exercise Overview

Global Citizen is one of the most innovative non-governmental organizations on the planet. It is the industry leader in philanthropic accountability and continues to grow as a true voice of the people. One of the major issues Global Citizen sees today is the lack of transparency and accountability surrounding a donation itself. If as a society we are going to problems such as extreme poverty or infectious disease, we need to make sure that every cause is getting the money promised, especially from the world's most impactful donor groups: federal governments.

Blockchain can provide the transparency and accountability that citizens demand. In this lab we walk you through how to build a simple three-member (Government, Aid Organization, and Global Citizen) network using the Hyperledger composer playground on which cause-specific pledges and fund transfers are made by the government, registered with aid organizations, and validated by Global Citizen.



Set up the Hyperledger Composer Playground

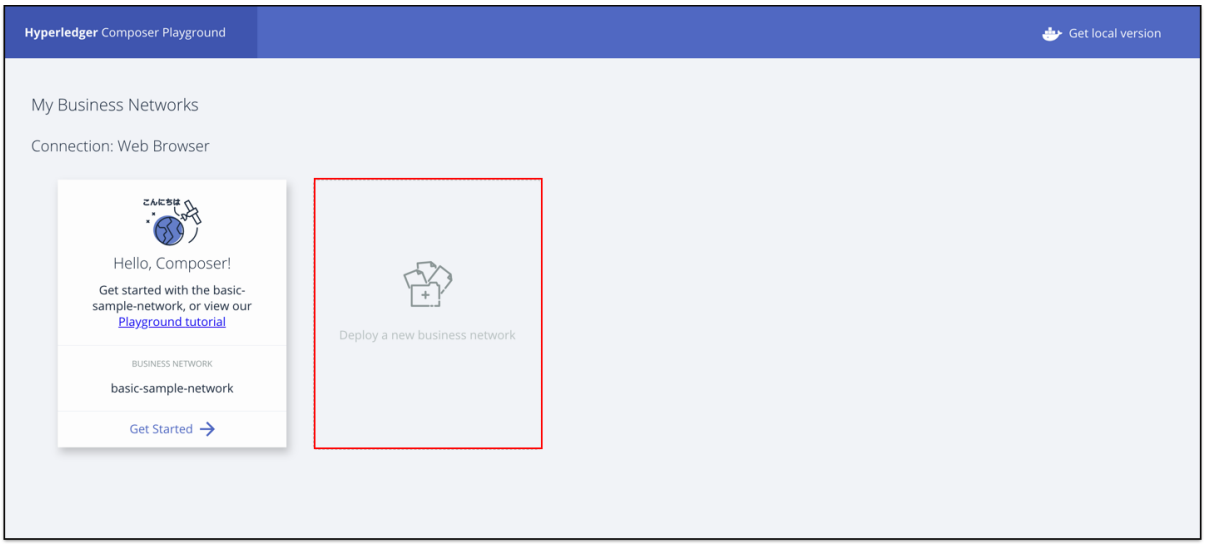
Hyperledger Composer is an open-source set of tools designed to make building blockchain applications easier. It allows users to model the business networks, assets and transactions that are required for blockchain applications, and to implement those transactions using simple JavaScript functions

This section of the lab takes place entirely in the web browser using the **Hyperledger Composer Playground**. Playground simulates the entire blockchain network within the browser by providing a sandpit environment to define, test and explore business networks defined using the Hyperledger Composer.

Note: Hyperledger Composer Playground is one method to use Hyperledger Composer, other methods are also available at www.fabric-composer.org.

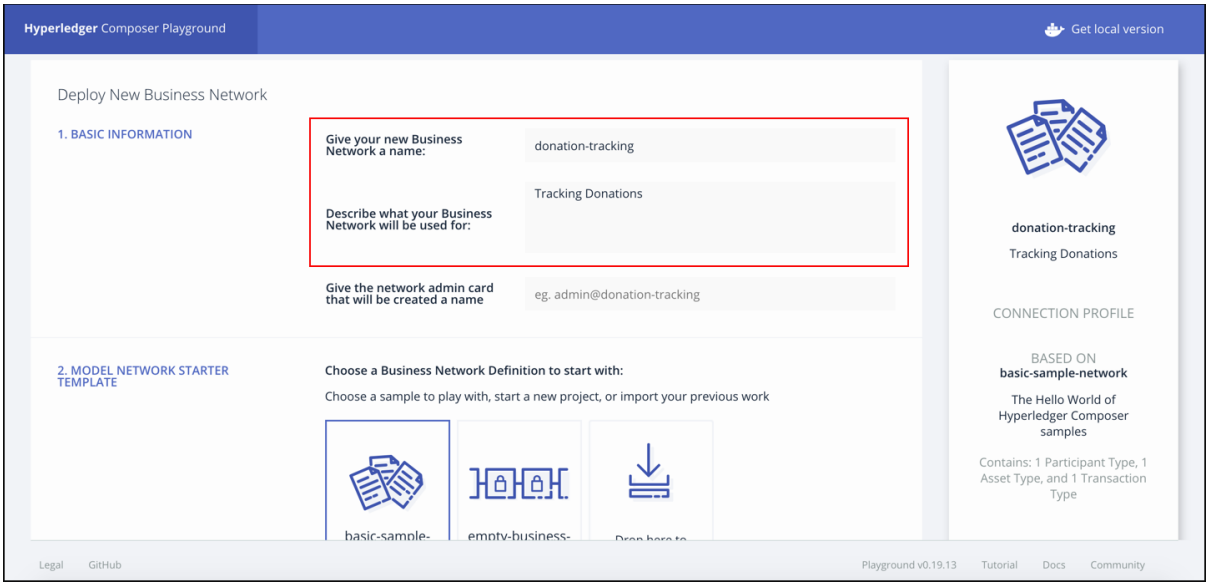
Now let's start building your blockchain application.

1. Open a web browser and go to http://composer-playground.mybluemix.net
2. Click on **Deploy a new business network**

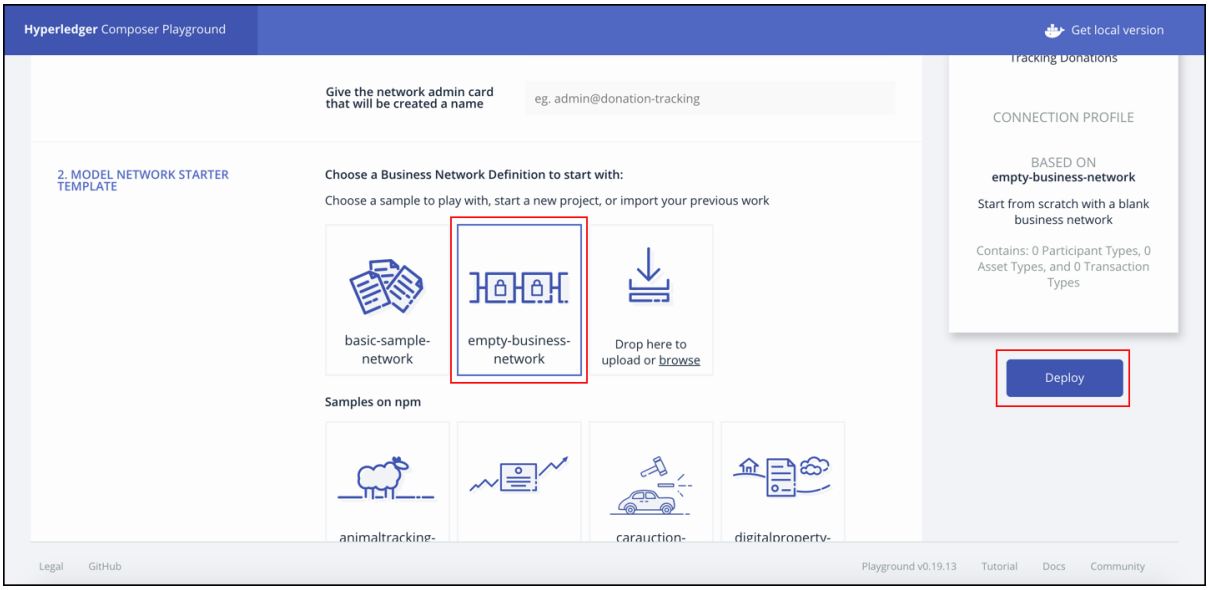


Note: By default, files are saved to local browser storage. If you have previously run this lab or edited files within this web page, then in order to run through this lab you will need to delete your browser cookies from the ***mybluemix.net*** domain.

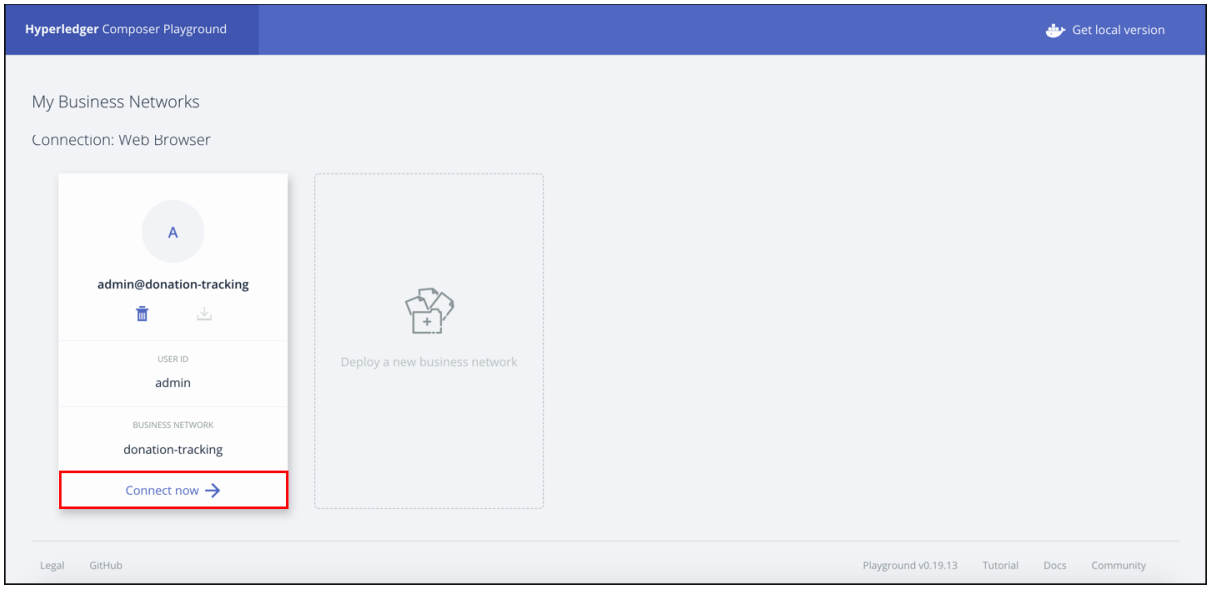
3. Give your application a suitable **Title** and **Description**



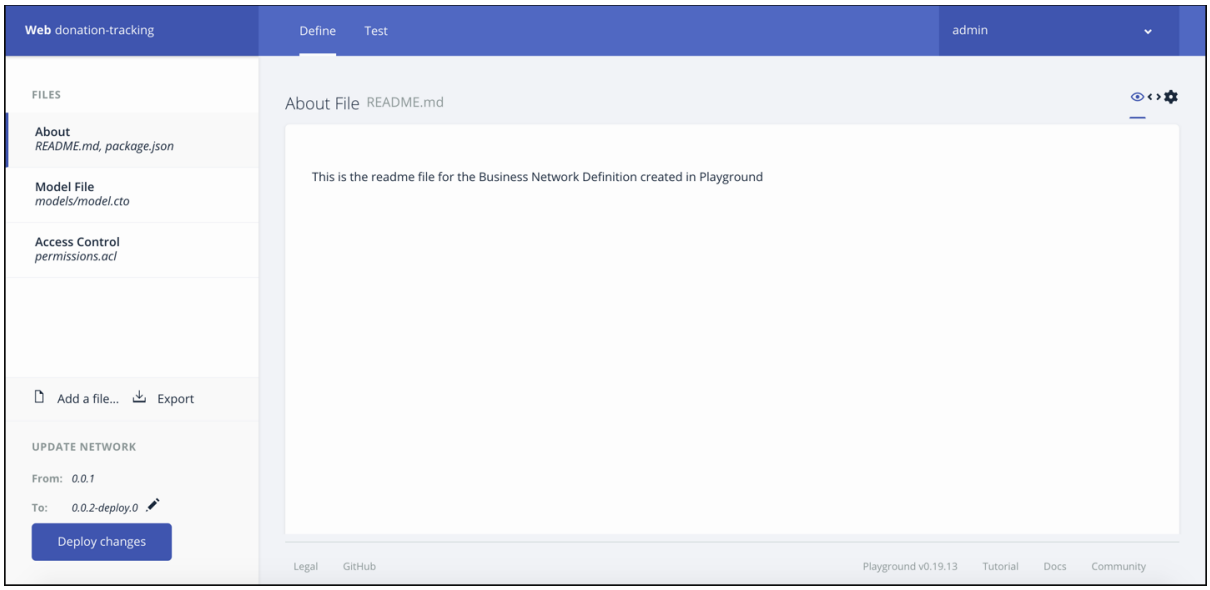
4. Select the **empty-business-network** and click on **Deploy**



5. Once the application has deployed, click on **Connect now**



6. If you see the page displayed below it means you have successfully created an instance of the blockchain application and can start editing it.



Business Network Definition

The Business Network Definition is a key concept of the Hyperledger Composer programming model.

Business Network Definition is composed of:

- The **Model file:** That is typically created by business analysts, as they define the structure and relationships between model elements: assets, participants and transactions.

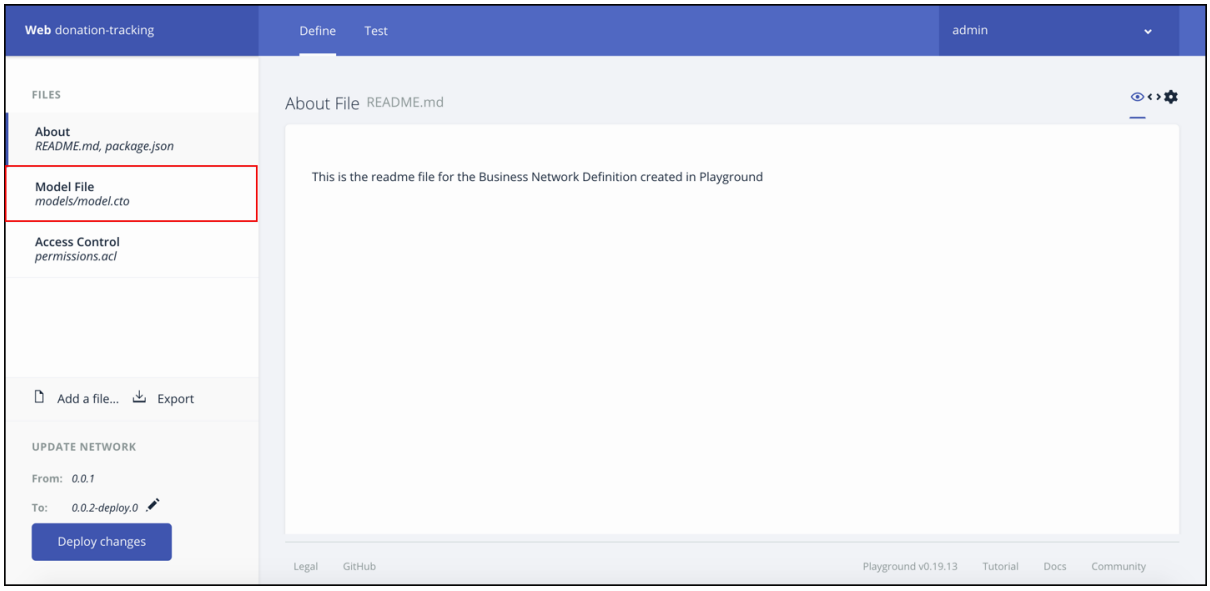
- The **Script file**: That is typically created by developers who are implementing business requirements provided by business analysts. It is the implementation of the blockchain application function, in our context, the tracking of donations.

- The **Access Control file:** Contains a set of access control rules that define the rights of the different participants in the business network.

In this section we define all the three files mentioned above.

Model File

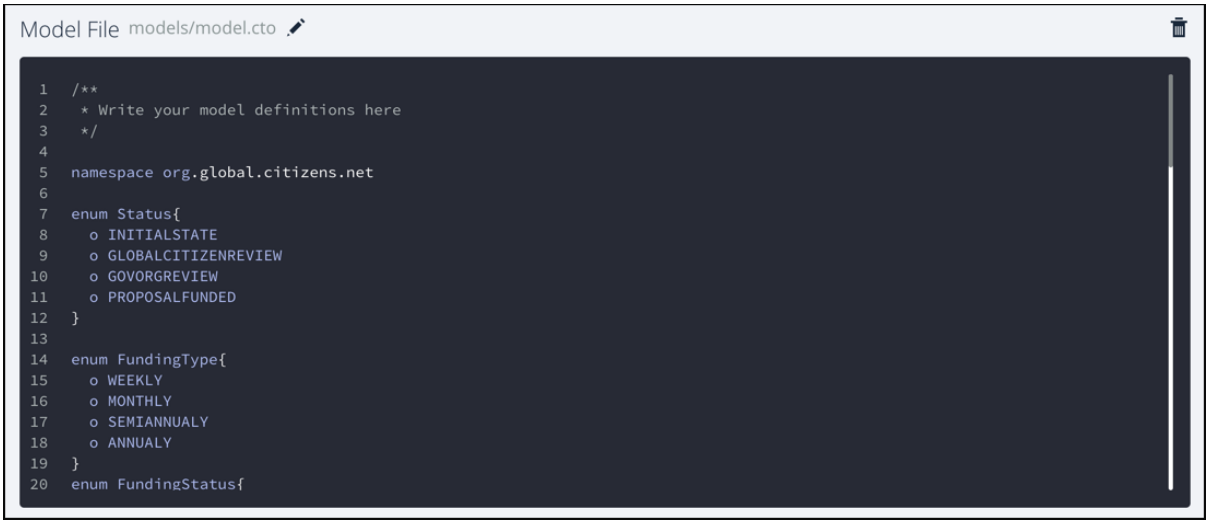
1. In order to create the Participant, click on **Model file**.



2. Once the model file opens, copy-paste the code snippet available here in the file.

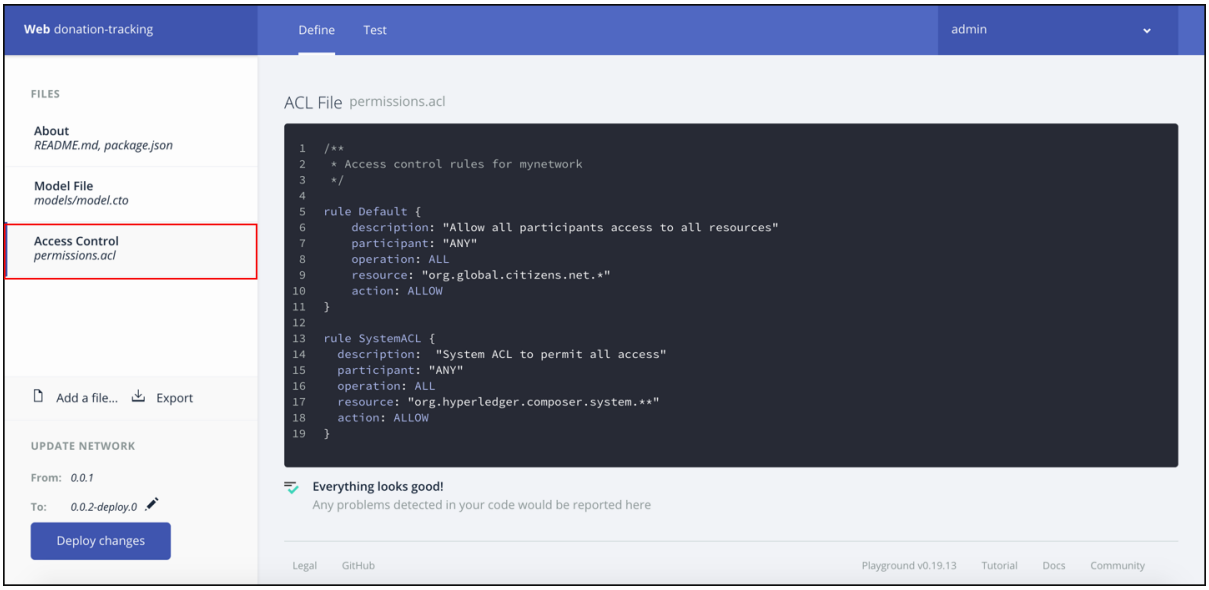
   In this file, we define 3 participants involved in the tracking of donation process: an organization representing a government entity (GovOrg), an organization representing on the provision of aid (AidOrg), and an organization representing Global Citizen (Global Citizen is an industry leader in philanthropic accountability and continues to grow as a true voice of the people).

   We also define 5 transactions to create a pledge and update them, as well as for transferring funds.



Access Control file

1. Click the **Access Control** (permissions.acl) file.



2. Once the access control file opens, copy-paste the code snippet available here in the file.

   In this file, access control rules will determine which users/roles are permitted to create, read, update or delete elements in a business network.

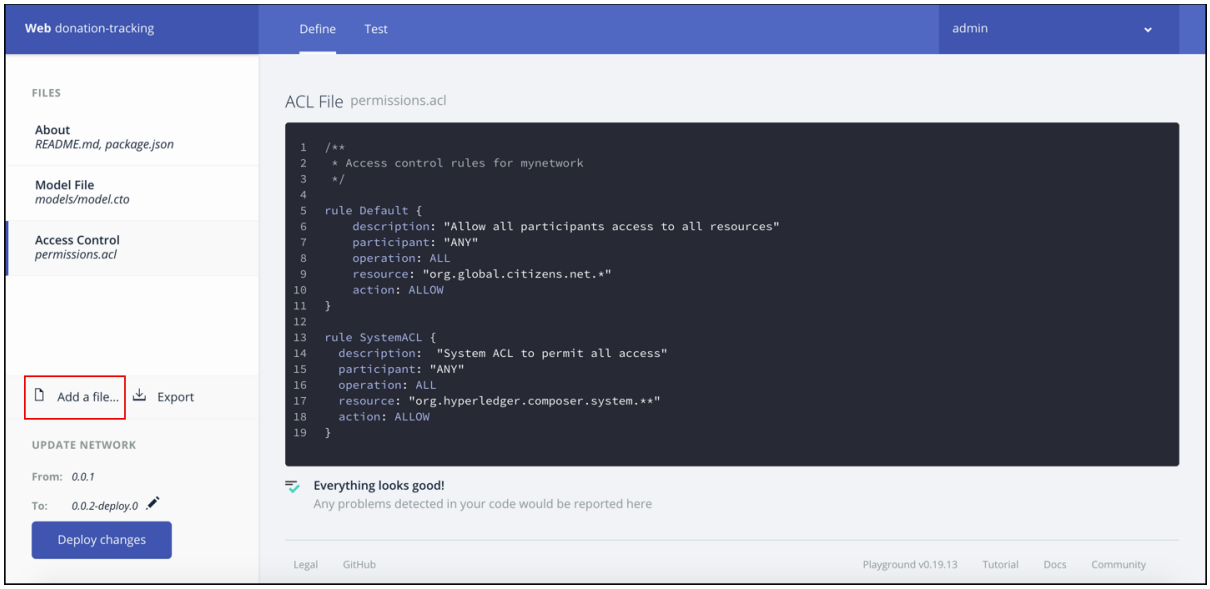
   In the context of this tracking blockchain application, all participants are allowed to access all resources and perform any of the above functions.

ACL File permissions.acl

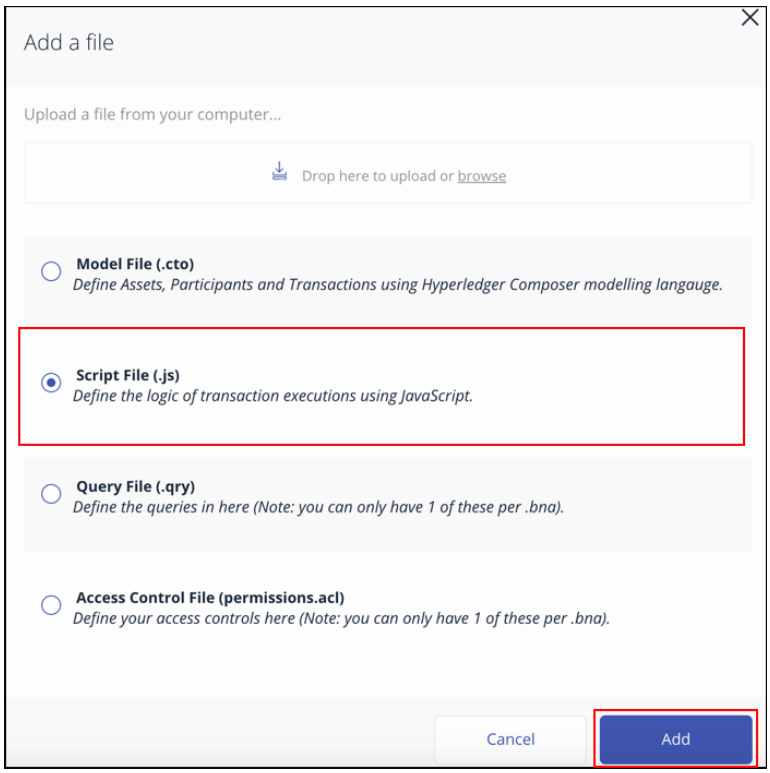
```
/**
 * Access control rules for mynetwork
 */

rule Default {
    description: "Allow all participants access to all resources"
    participant: "ANY"
    operation: ALL
    resource: "org.global.citizens.net.*"
    action: ALLOW
}

rule SystemACL {
  description:  "System ACL to permit all access"
  participant: "ANY"
  operation: ALL
  resource: "org.hyperledger.composer.system.**"
  action: ALLOW
}
```

Script file

1. Click **Add a file**

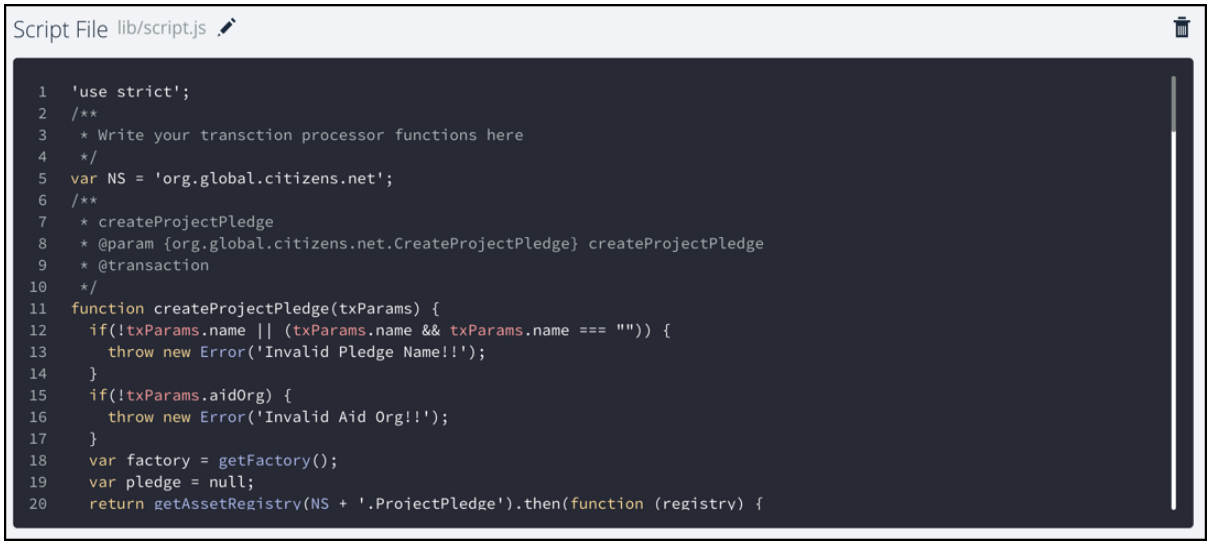


2. Select **Script File** and click **Add.**

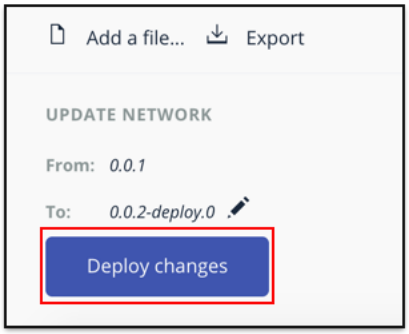


3. Once the Script File opens, copy-paste the code snippet [available here] in the file.

   This file contains transaction processor functions that contains the rules of tracking the donations using this Blockchain application.



4. After adding all the code snippets to the Model, Script and ACL files, click **Deploy Changes**

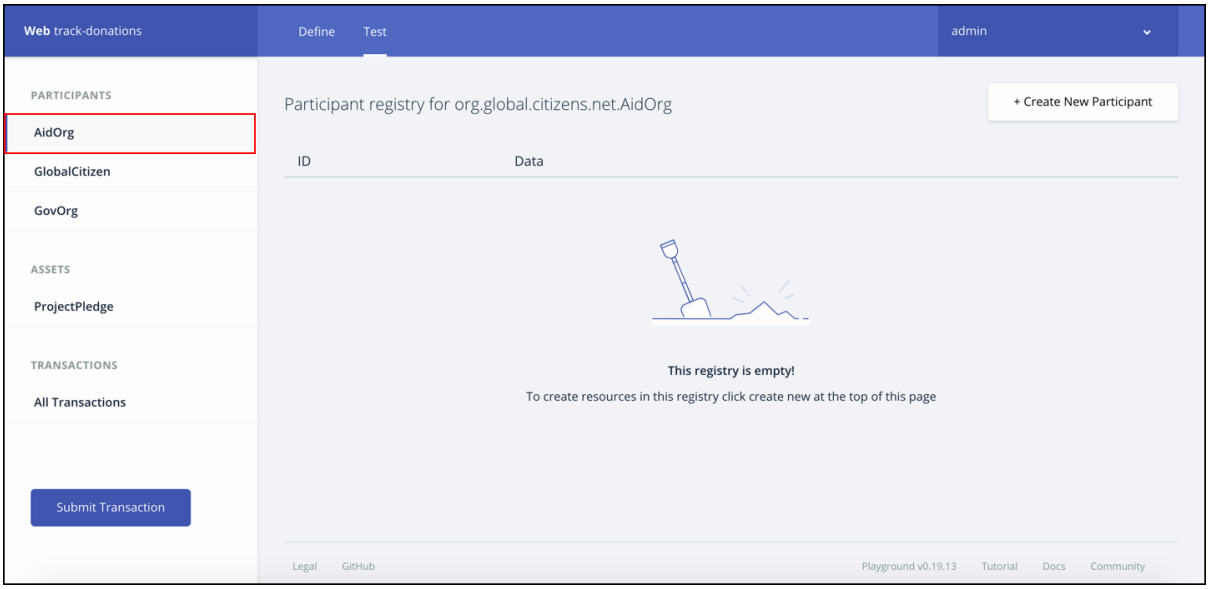


Add Participants

In this section, we will create instances of all the three participants (Government, Aid Organization, and Global Citizen) associated with this blockchain application.

1. Click the **Test** tab.

2. Click on **AidOrg** to view that there are no Aid Organizations added in the environment.

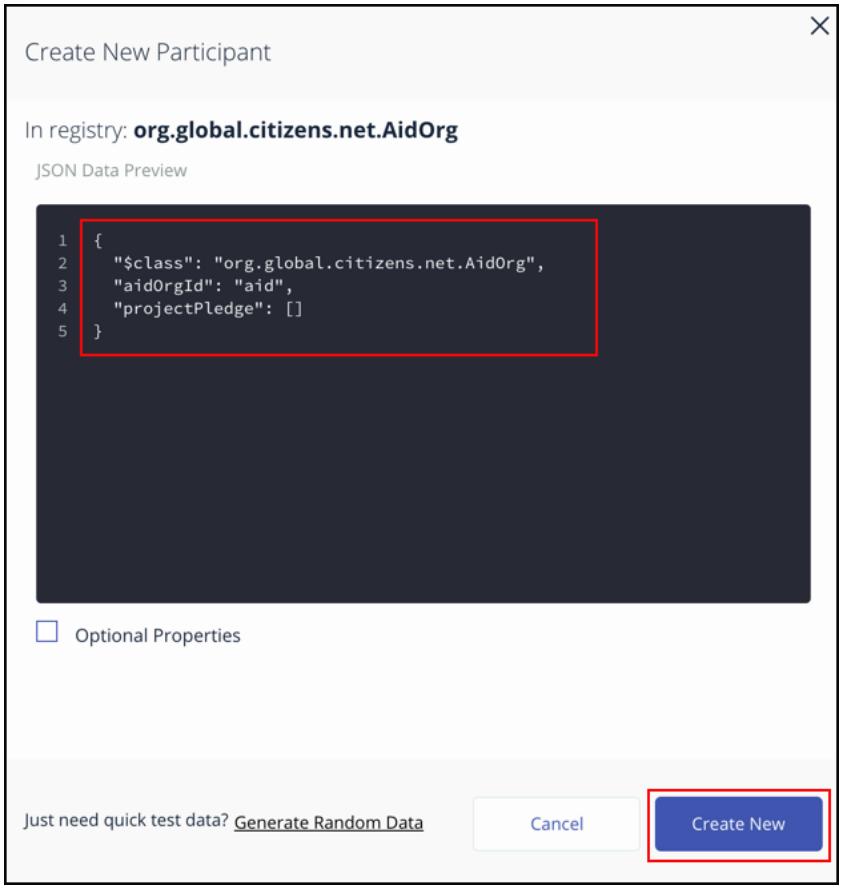


3. Click **Create New Participant** to add a new Aid Organizations.

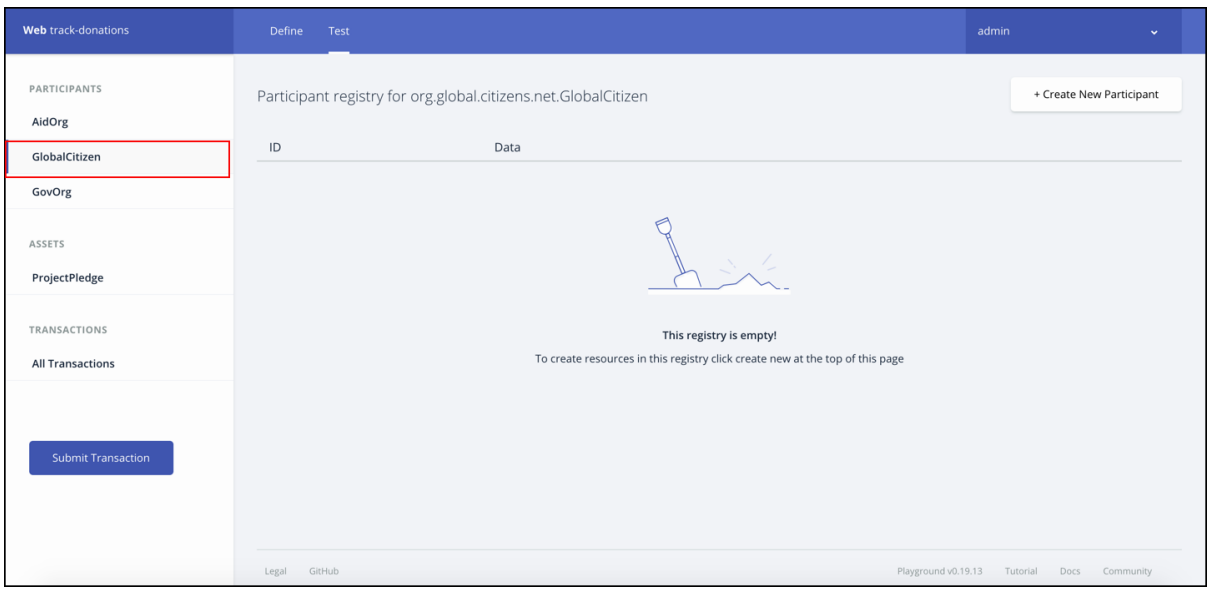
4. Type the values given below into the JSON data structure, and click **Create New**

```
{
  "$class": "org.global.citizens.net.AidOrg",
```

```
  "aidOrgId": "aid",
  "projectPledge": []
}
```



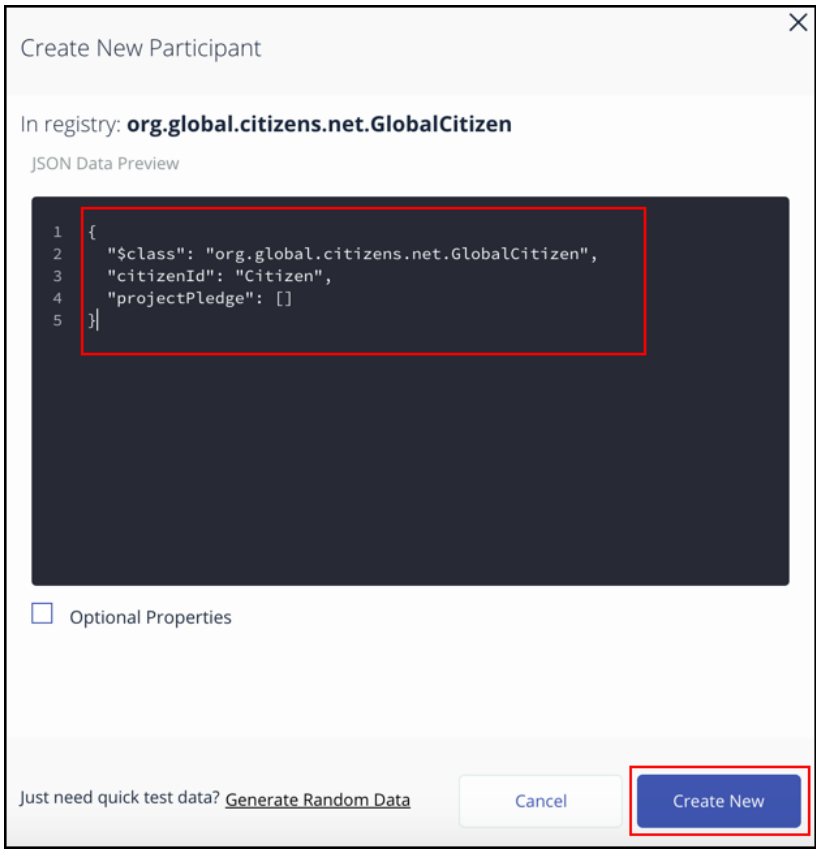
5. Click on **GlobalCitizen** to view that there are no citizens added in the environment.



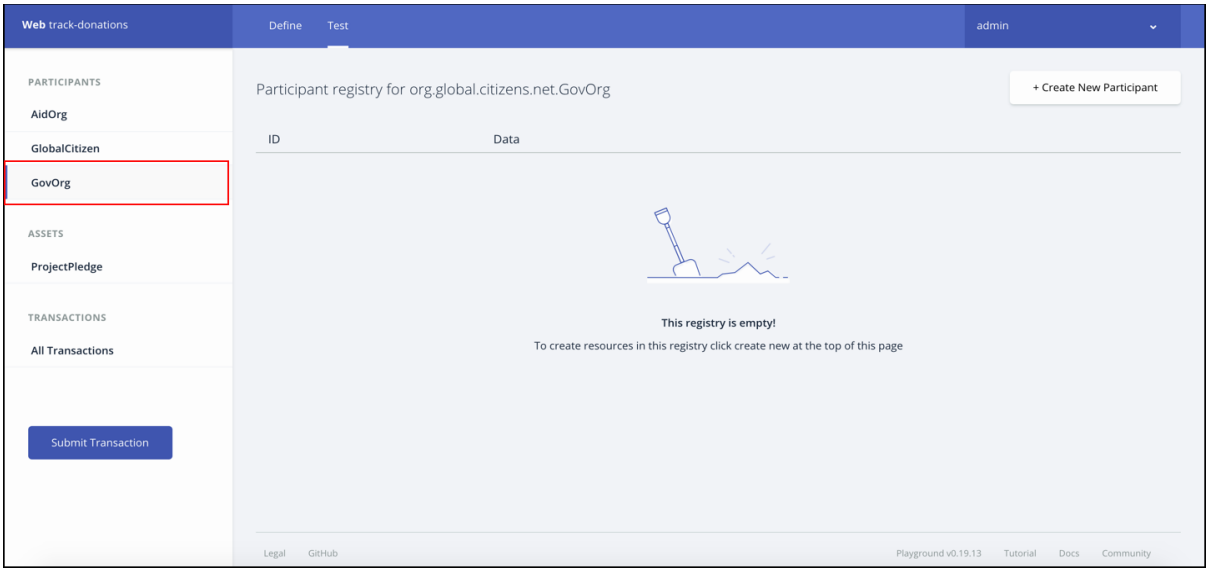
6. Click **Create New Participant**

7. Type the values given below into the JSON data structure, and click **Create New**

```
{
  "$class":"org.global.citizens.net.GlobalCitizen",
  "citizenId": "Citizen",
  "projectPledge": []
}
```



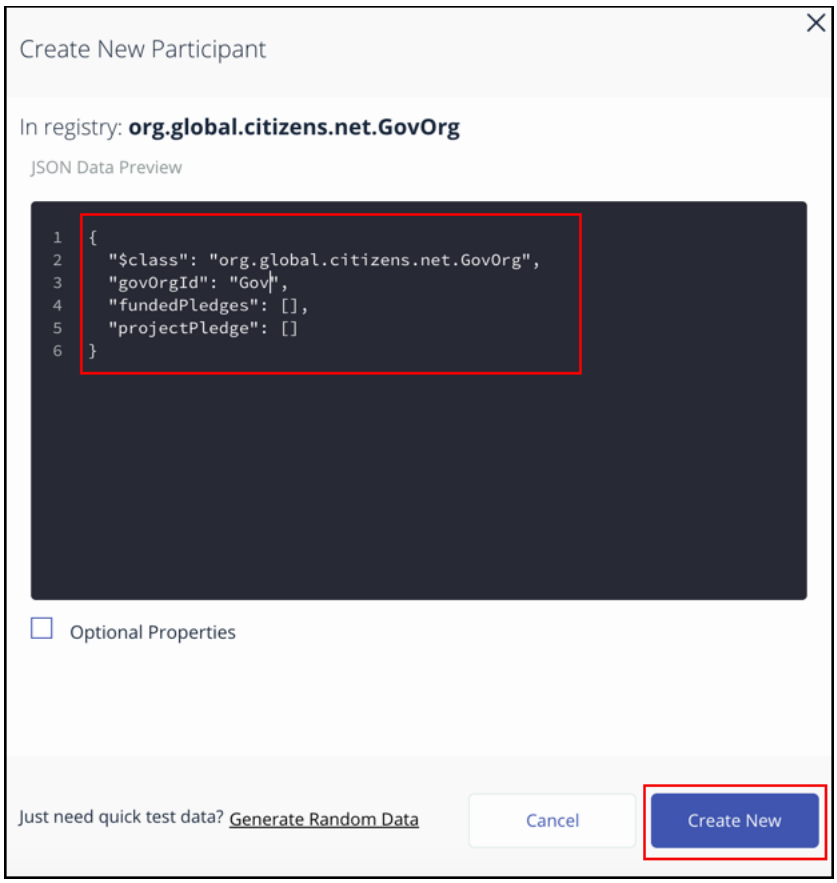
8. Click on **GovOrg** to view that there are no Government Organizations added in the environment.



9. Click **Create New Participant**

10. Type the values given below into the JSON data structure

```
{
  "$class": "org.global.citizens.net.GovOrg",
  "govOrgId": "Gov",
  "fundedPledges": [],
  "projectPledge": []
}
```

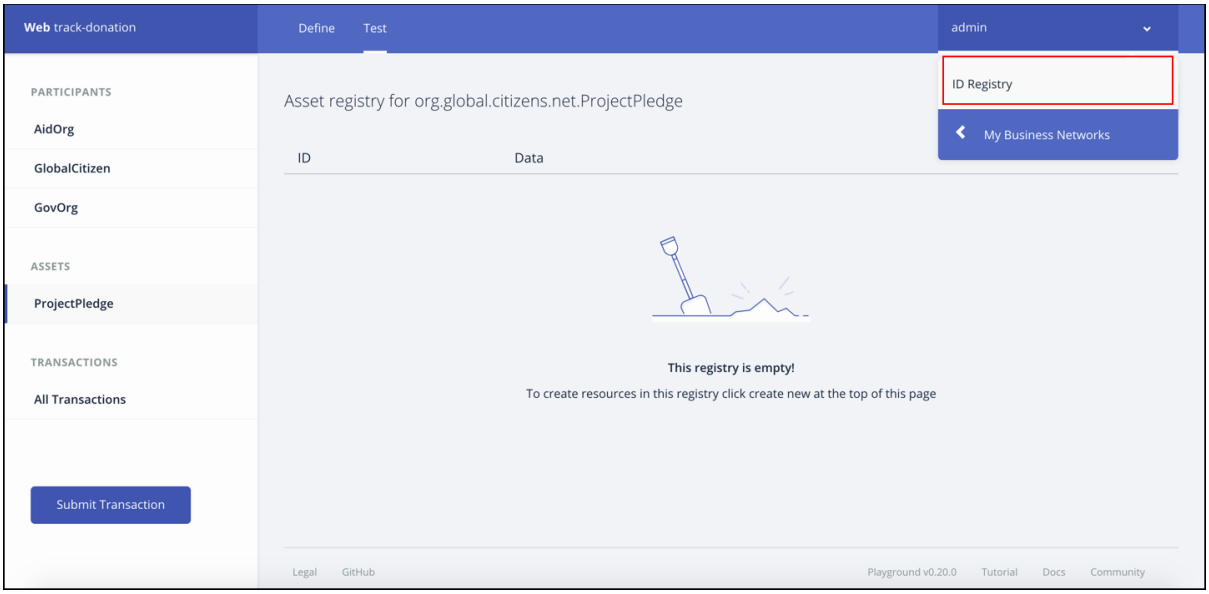


Add Network Cards

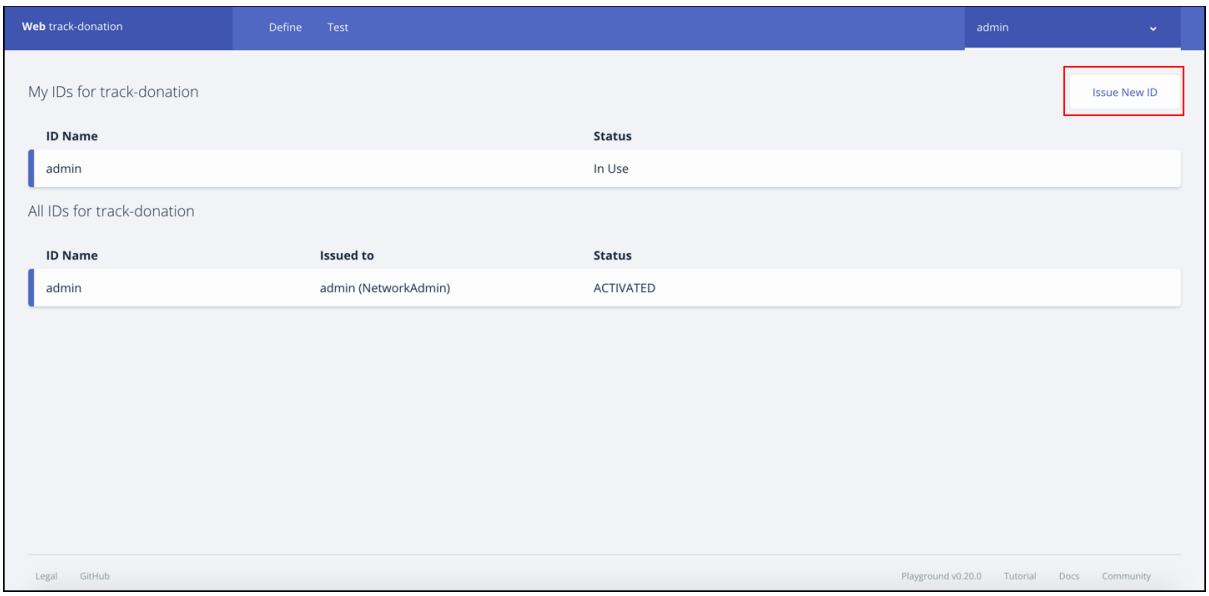
Connection profiles contain the information necessary to connect to a fabric. **Business Network Cards** combine a connection profile, identity, and certificates to allow a connection to a business network in Hyperledger Composer Playground.

In order to add the Business Network Cards, you need to follow the steps below:

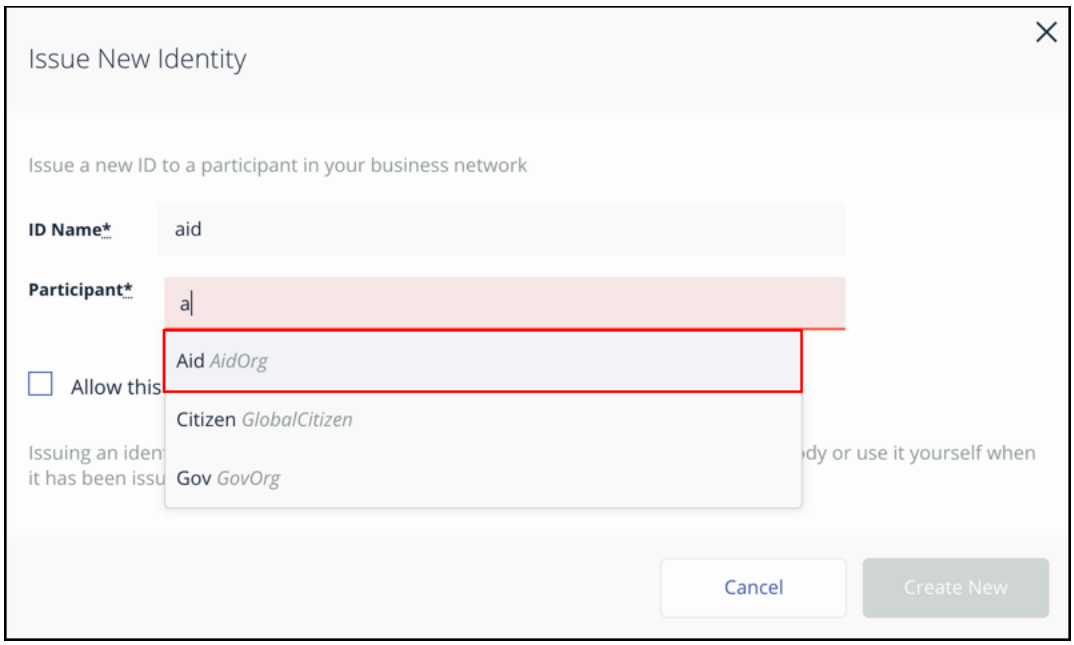
1. Click **ID Registry**



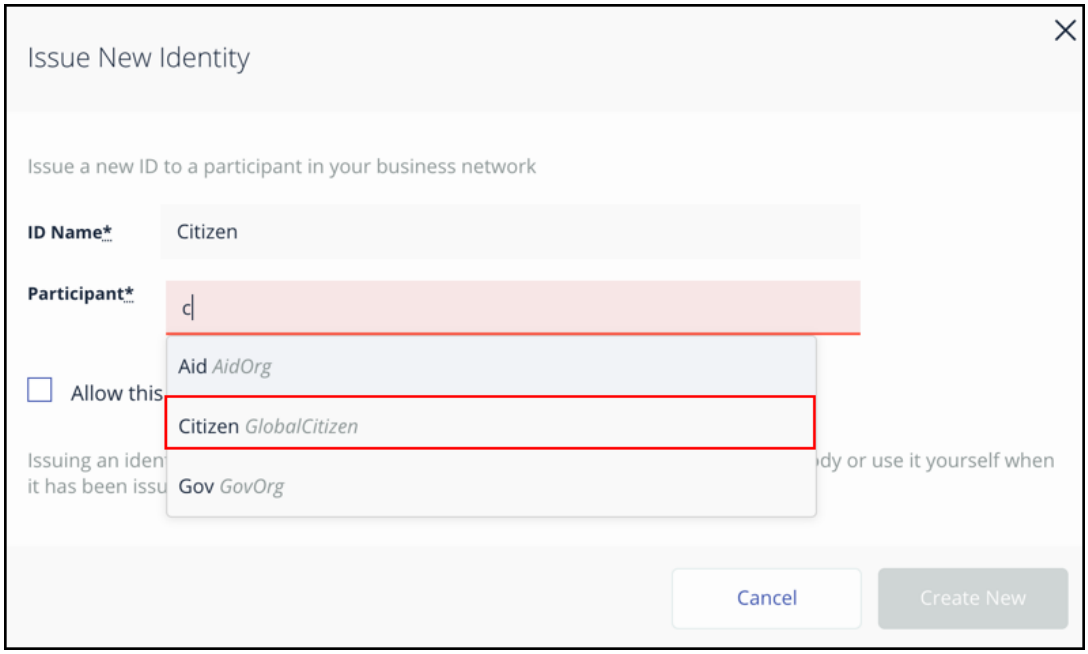
2. Click **Issue New ID**



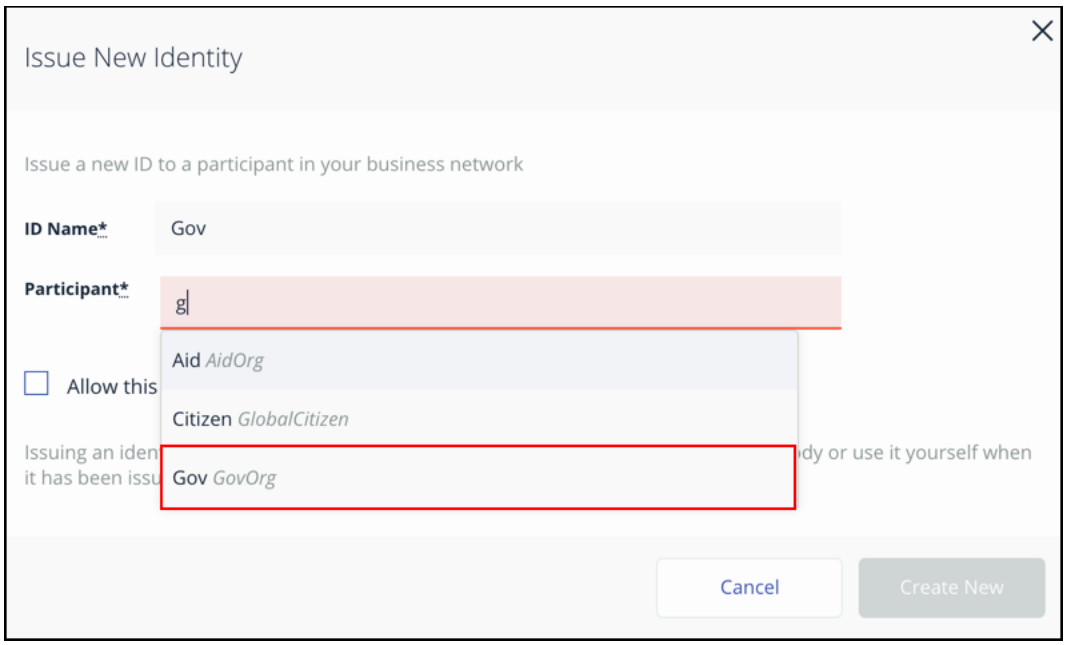
3. Assign the IDs



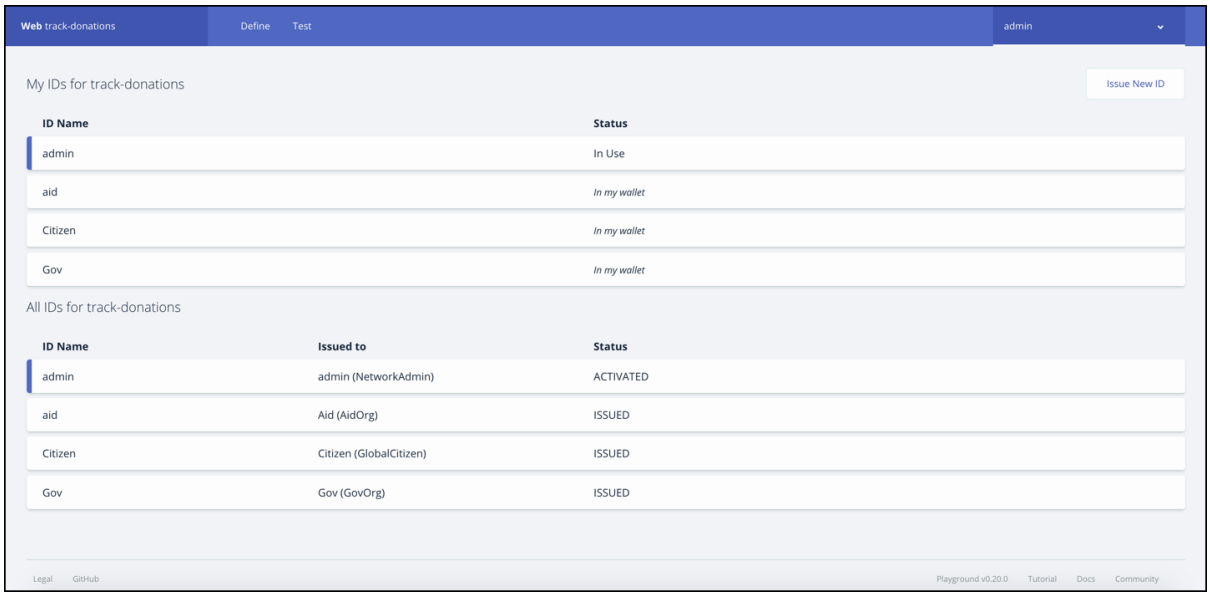
4. Assign the ID and click create new



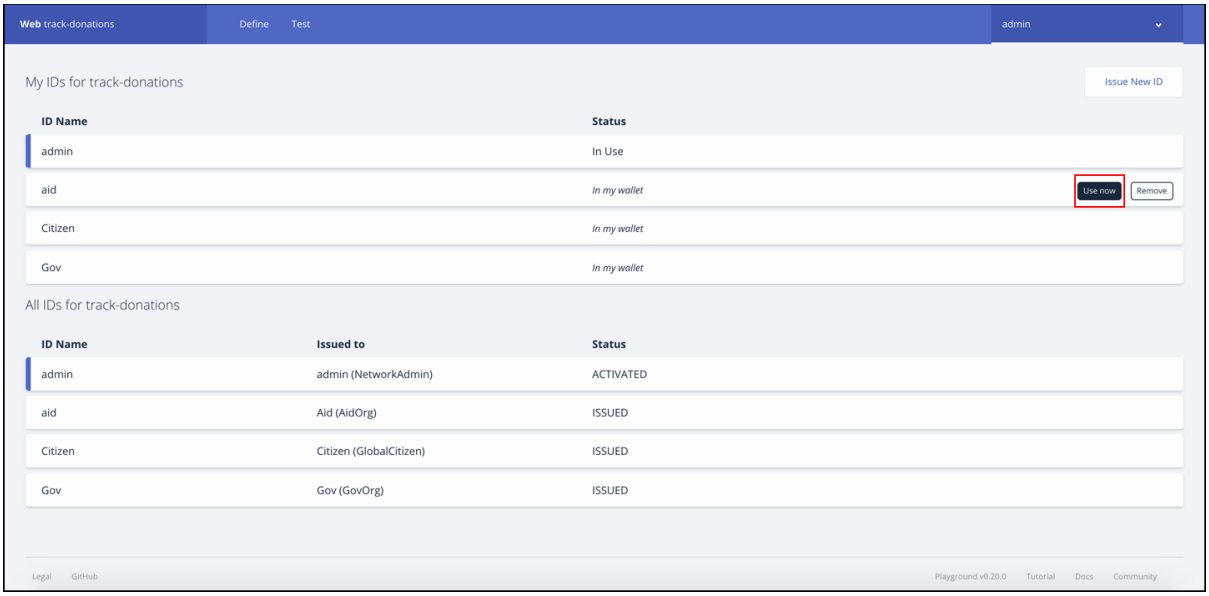
5. Assign the ID and click create new



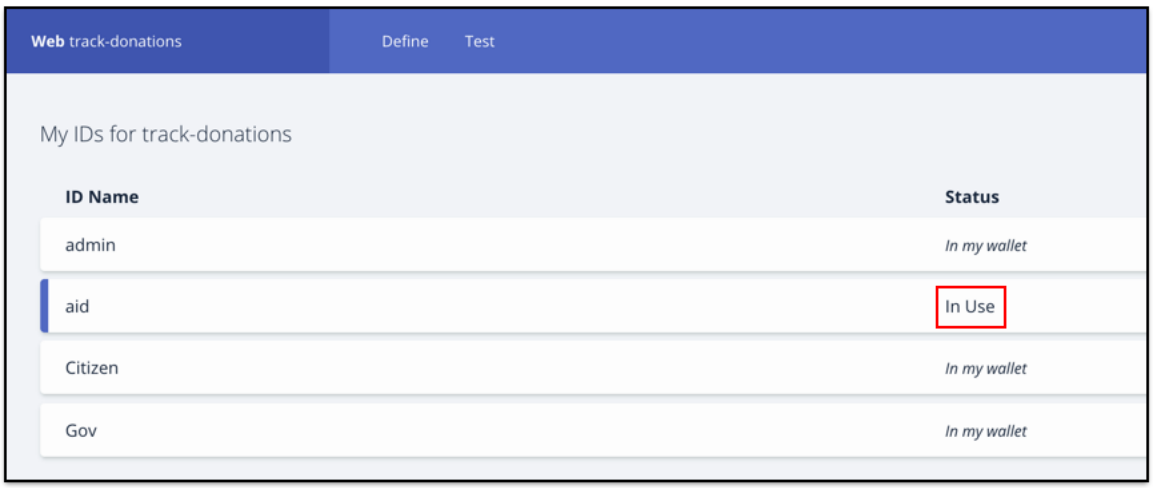
6. Once the ID's are created you will be able to view then in the ID Registry



7. Click on **Use Now** to select the **AidOrg** participant registry to perform transactions on network.



8. You will be able to change in the status once you click on **Use Now**

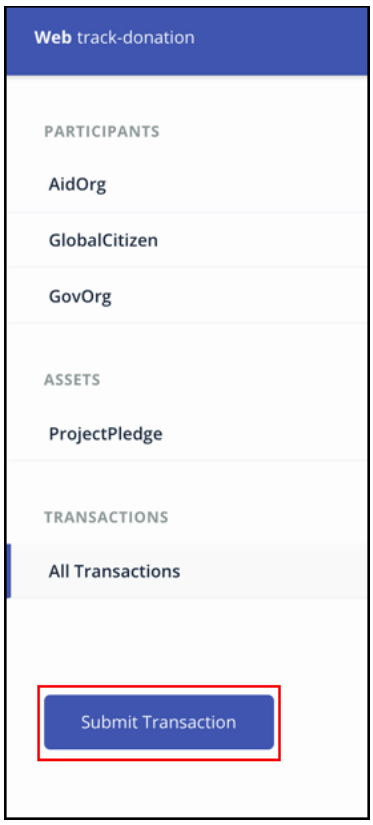


Submit your Transactions

After creating the **Network Cards,** you can start testing the Blockchain network by performing transactions.

1. Click the **Test** tab.

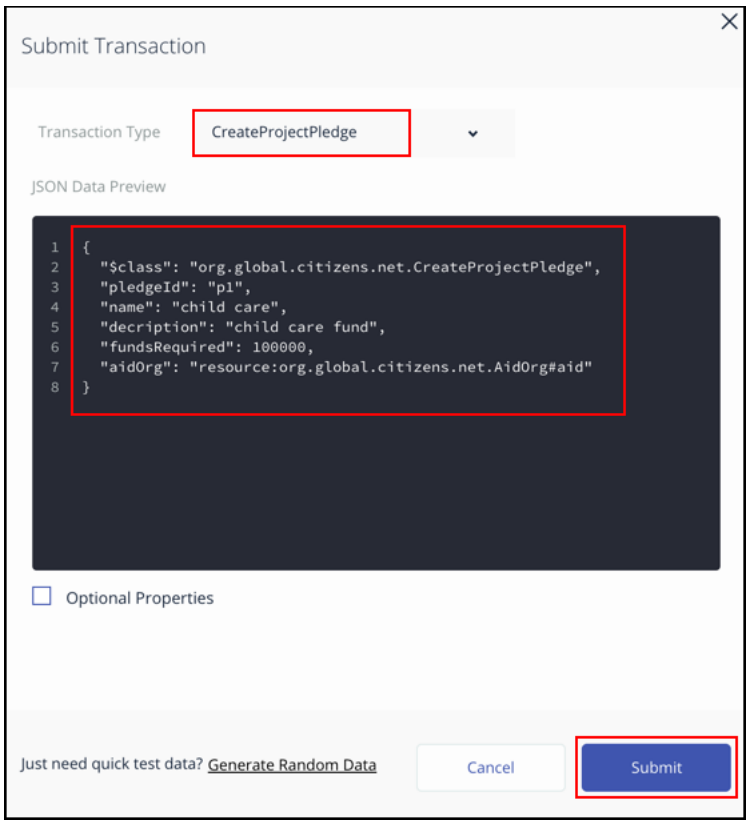
2. Click **Submit Transaction** to create a new transaction.



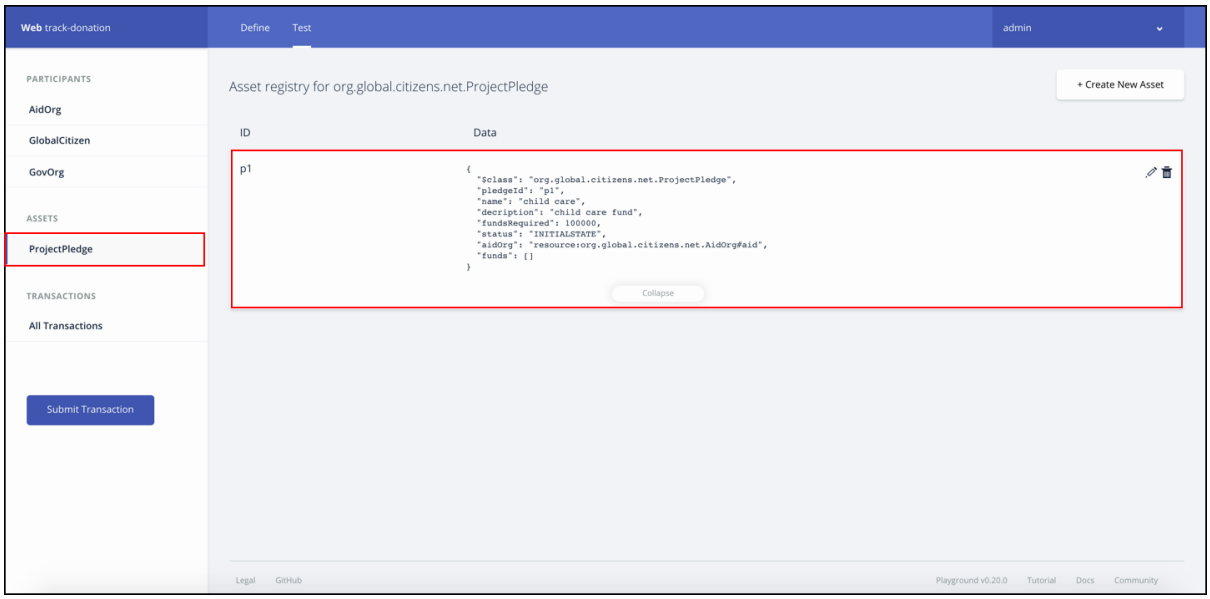
3. From the Transaction Type select **CreateProjectPledge** to create an instance of the charitable cause for which funds need to be collected.

   Add the values given below into the JSON data structure. Once done, click **Submit**.

```
{
  "$class":
"org.global.citizens.net.CreateProjectPledge",
  "pledgeId": "p1",
  "name": "child care",
  "decription": "child care fund",
  "fundsRequired": 100000,
  "aidOrg": "resource:org.global.citizens.net.AidOrg#aid"
}
```



4. Click **ProjectPledge** to view the pledge you have created.

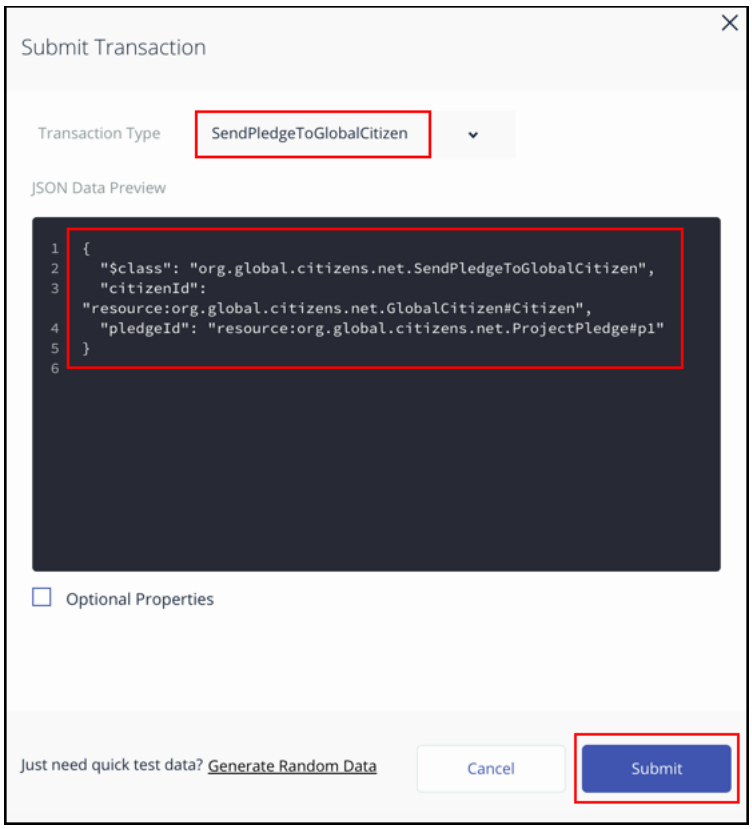


5. After creating the pledge, it will be shown in the **AidOrg**.

6. Once the pledge has been created, the aid organization will send the pledge proposal to Global Citizen organization for review.

   In order to do this, you have to create a new transaction. Select **SendPledgeToGlobalCitizen** as the transaction type and add the values given below into the JSON data structure. Once done, click **Submit**.

```
{
  "$class":
"org.global.citizens.net.SendPledgeToGlobalCitizen",
  "citizenId":
"resource:org.global.citizens.net.GlobalCitizen#Citizen",
  "pledgeId":
"resource:org.global.citizens.net.ProjectPledge#p1"
}
```



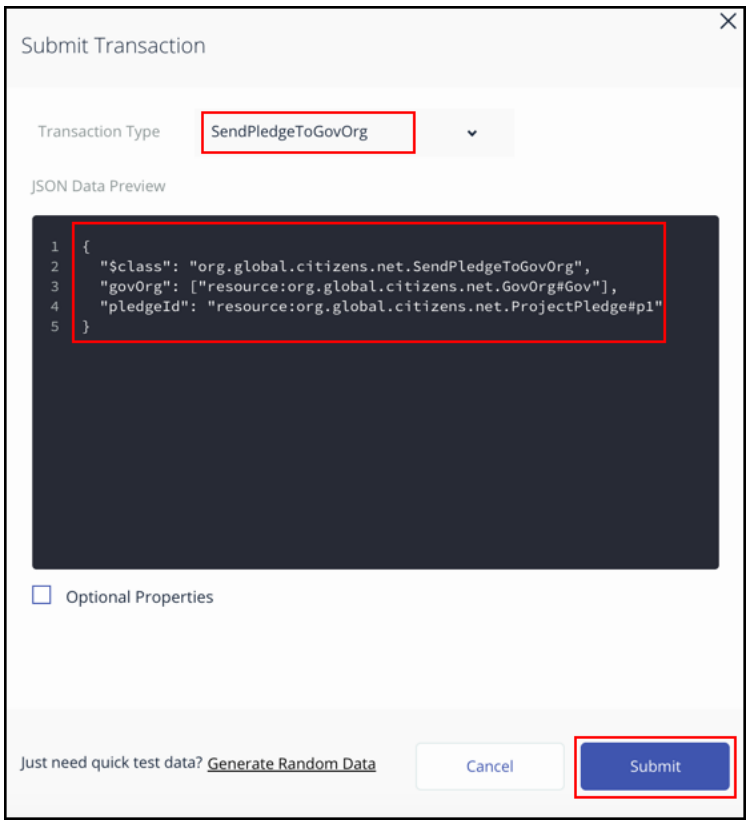
7. Once you submit the transaction, the Global Citizen participant registry will get updated with the new pledge request.

8. After successful verification; Global Citizen submits a **SendPledgeToGovOrg** transaction to get funds for the project pledge from government organizations.

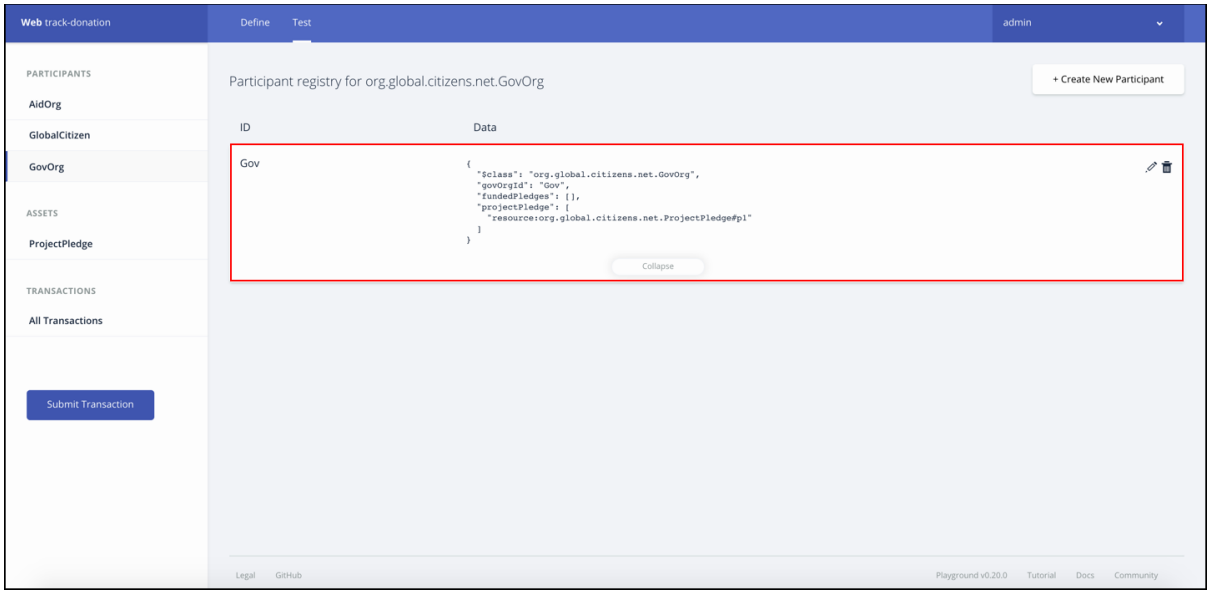
   Add the values given below into the JSON data structure. Once done, click **Submit**.

   {

```
  "$class": "org.global.citizens.net.SendPledgeToGovOrg",
  "govOrg":
["resource:org.global.citizens.net.GovOrg#gov"],
  "pledgeId":
"resource:org.global.citizens.net.ProjectPledge#p1"
}
```



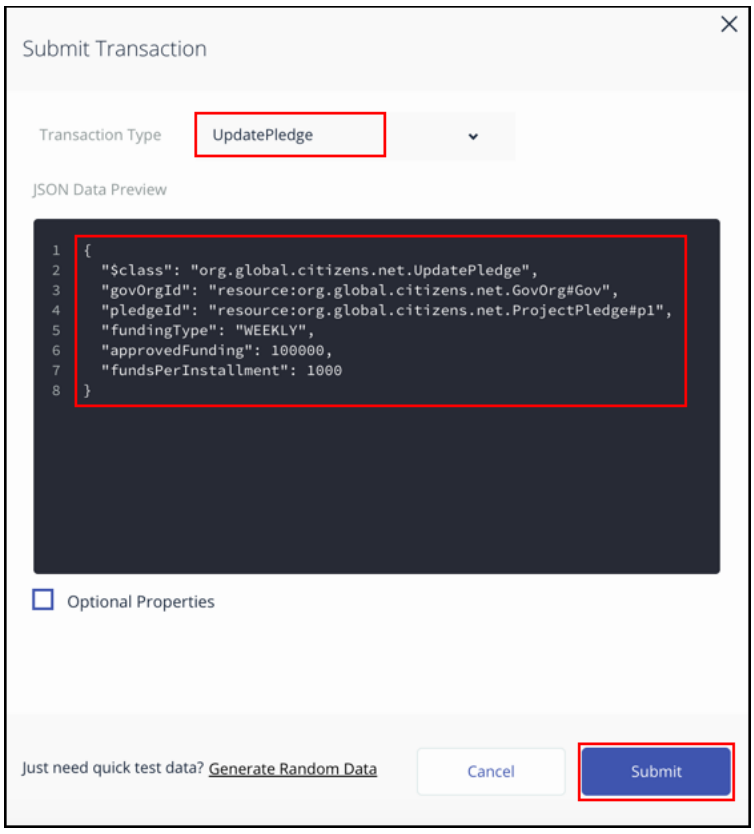
9. After sending the pledge to the Government Organization, it will be shown in the **GovOrg**.



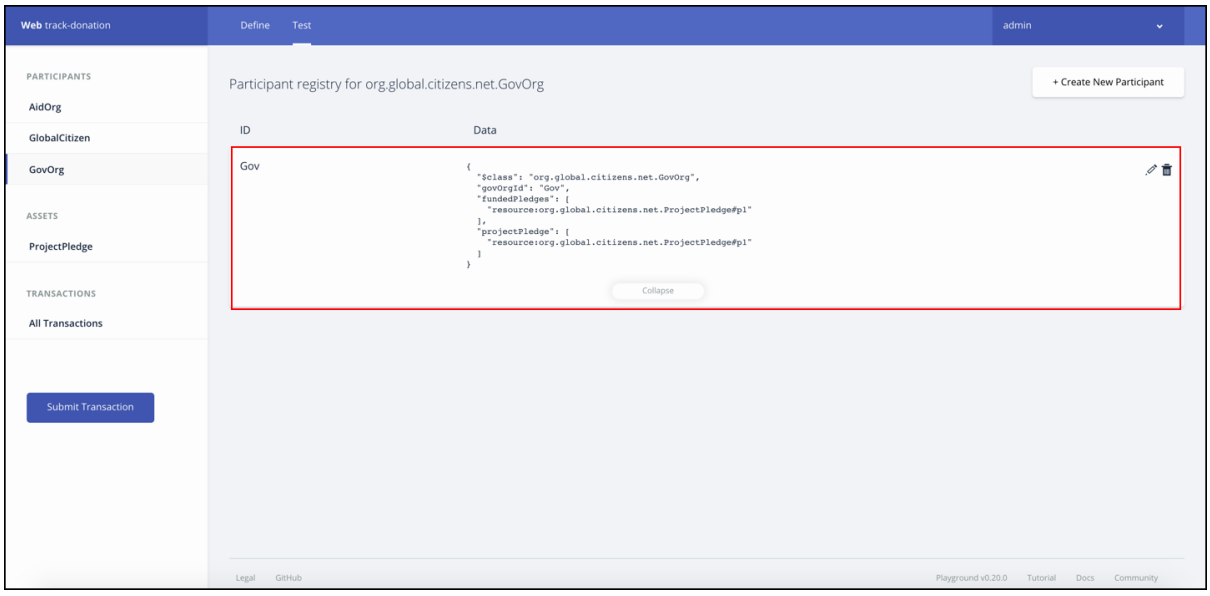
10. After reviewing if the Government decides to fund the project they submit a **UpdatePledge** transaction to update the project pledge asset.

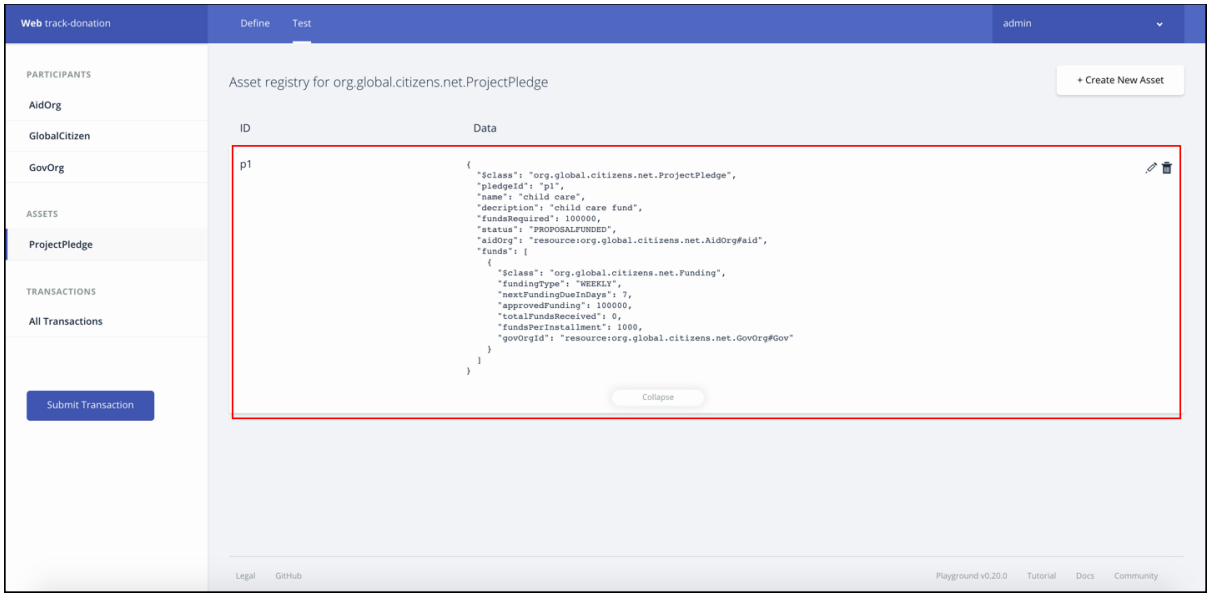
Add the values given below into the JSON data structure. Once done, click **Submit**.

```
{
  "$class": "org.global.citizens.net.UpdatePledge",
                                                "govOrgId":
"resource:org.global.citizens.net.GovOrg#Gov",
                                                "pledgeId":
"resource:org.global.citizens.net.ProjectPledge#p1",
  "fundingType": "WEEKLY",
  "approvedFunding": 100000,
  "fundsPerInstallment": 1000
}
```



11. The Update to the pledge will be reflected in **GovOrg** and in the **ProjectPledge**

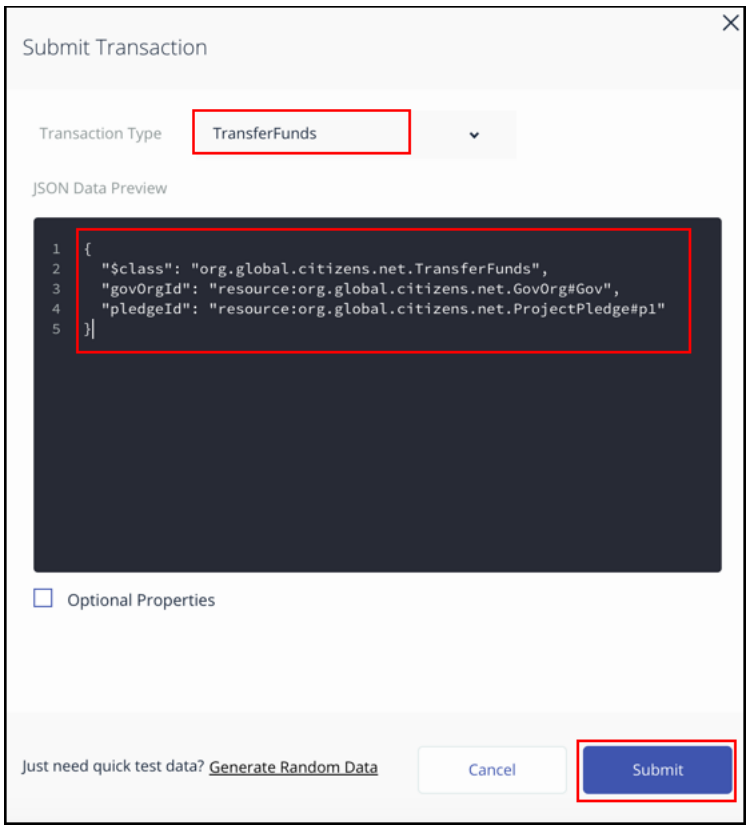




12. Government organizations periodically sends the funds to project by submitting **TransferFunds** transaction.

Add the values given below into the JSON data structure. Once done, click **Submit**.

```
{
  "$class": "org.global.citizens.net.TransferFunds",
                                                    "govOrgId":
"resource:org.global.citizens.net.GovOrg#Gov",
                                                    "pledgeId":
"resource:org.global.citizens.net.ProjectPledge#p1"
}
```



13. The transfer of funds is reflected in the **ProjectPledge**