

Track donations with Blockchain

You can find the code at:

<https://github.com/AreejEssa/global-citizen>

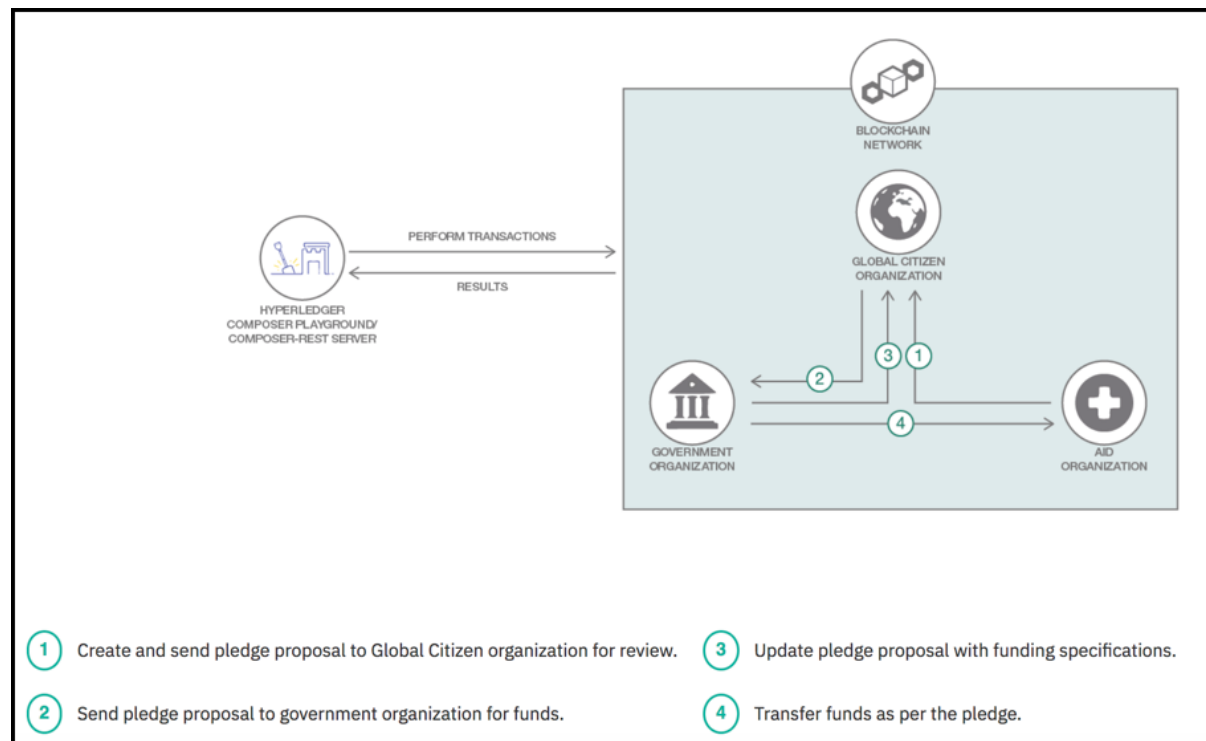
(in the “BC-Workshop” folder – first folder in the repo)

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

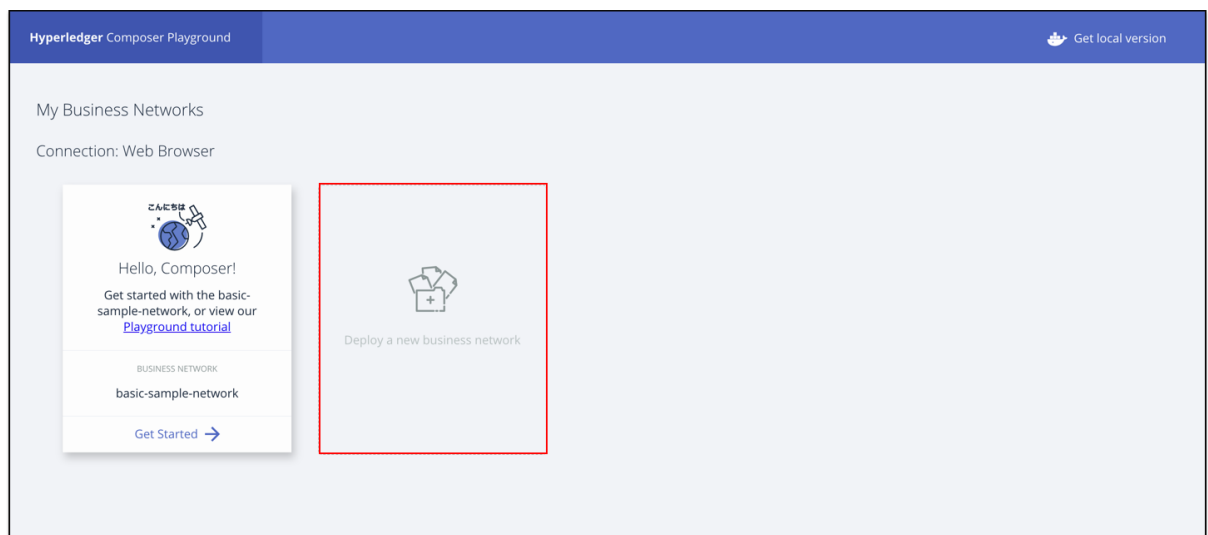
[Hyperledger Composer](#) is an open-source set of tools designed to make building blockchain applications easier. It allows users to model the business networks, assets and transactions that are required for blockchain applications, and to implement those transactions using simple JavaScript functions

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**

Hyperledger Composer Playground

Get local version

Deploy New Business Network

1. BASIC INFORMATION

Give your new Business Network a name:

donation-tracking

Describe what your Business Network will be used for:

Tracking Donations

Give the network admin card that will be created a name

eg. admin@donation-tracking

2. MODEL NETWORK STARTER TEMPLATE

Choose a Business Network Definition to start with:

Choose a sample to play with, start a new project, or import your previous work

basic-sample-network

empty-business-network

Drop here to upload or browse

donation-tracking

Tracking Donations

CONNECTION PROFILE

BASED ON

basic-sample-network

The Hello World of Hyperledger Composer samples

Contains: 1 Participant Type, 1 Asset Type, and 1 Transaction Type

Legal GitHub

Playground v0.19.13 Tutorial Docs Community

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

Hyperledger Composer Playground

Get local version

Tracking Donations

CONNECTION PROFILE

BASED ON

empty-business-network

Start from scratch with a blank business network

Contains: 0 Participant Types, 0 Asset Types, and 0 Transaction Types

Deploy

2. MODEL NETWORK STARTER TEMPLATE

Choose a Business Network Definition to start with:

Choose a sample to play with, start a new project, or import your previous work

basic-sample-network

empty-business-network

Drop here to upload or browse

Samples on npm

animaltracking-

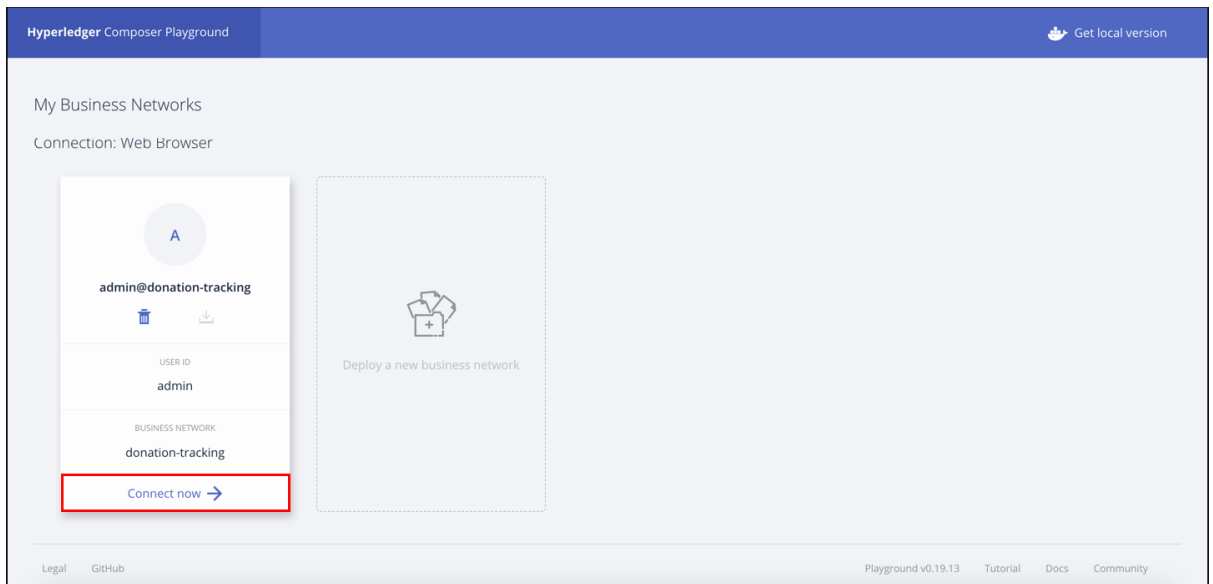
carauction-

digitalproperty-

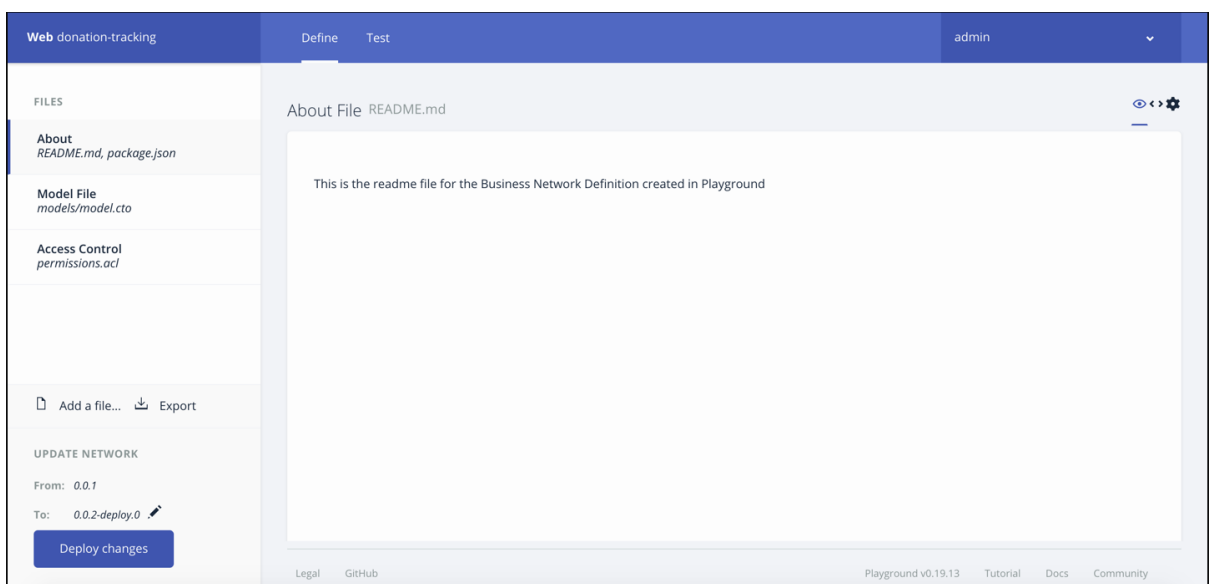
Legal GitHub

Playground v0.19.13 Tutorial Docs Community

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:

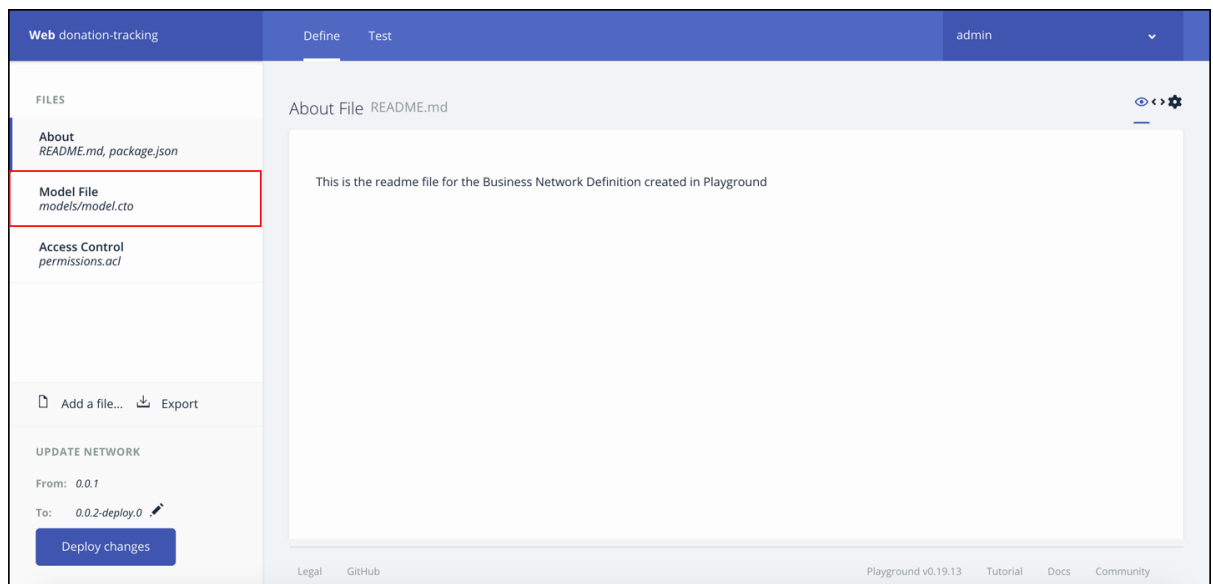
- The **Model file**: That is typically created by business analysts, as they define the structure and relationships between model elements: assets, participants and transactions.
- The **Script file**: That is typically created by developers who are implementing business requirements provided by business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

```
Model File models/model.cto

1  /**
2   * Write your model definitions here
3   */
4
5   namespace org.global.citizens.net
6
7   enum Status{
8       o INITIALSTATE
9       o GLOBALCITIZENREVIEW
10      o GOVORGREVIEW
11      o PROPOSALFUNDED
12  }
13
14  enum FundingType{
15      o WEEKLY
16      o MONTHLY
17      o SEMIANNUALY
18      o ANNUALY
19  }
20  enum FundingStatusf
```

Access Control file

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

The screenshot shows the Hyperledger Composer Playground interface. On the left, a sidebar lists files: 'About' (README.md, package.json), 'Model File' (models/model.cto), and 'Access Control' (permissions.acl), which is highlighted with a red box. Below the sidebar are options to 'Add a file...' or 'Export', and a section for 'UPDATE NETWORK' showing a deployment from version 0.0.1 to 0.0.2-deploy.0 with a 'Deploy changes' button. The main area is titled 'Define' and shows the 'ACL File permissions.acl' with the following code:

```
1  /**
2   * Access control rules for mynetwork
3   */
4
5   rule Default {
6       description: "Allow all participants access to all resources"
7       participant: "ANY"
8       operation: ALL
9       resource: "org.global.citizens.net.*"
10      action: ALLOW
11  }
12
13  rule SystemACL {
14      description: "System ACL to permit all access"
15      participant: "ANY"
16      operation: ALL
17      resource: "org.hyperledger.composer.system.*"
18      action: ALLOW
19  }
```

Below the code, a green checkmark icon and the text 'Everything looks good!' are displayed, followed by the message 'Any problems detected in your code would be reported here'. At the bottom of the interface, there are links for 'Legal', 'GitHub', 'Playground v0.19.13', 'Tutorial', 'Docs', and 'Community'.

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.ac1
1  /**
2   * Access control rules for mynetwork
3   */
4
5  rule Default {
6    description: "Allow all participants access to all resources"
7    participant: "ANY"
8    operation: ALL
9    resource: "org.global.citizens.net.*"
10   action: ALLOW
11  }
12
13  rule SystemACL {
14    description: "System ACL to permit all access"
15    participant: "ANY"
16    operation: ALL
17    resource: "org.hyperledger.composer.system.*"
18    action: ALLOW
19  }
```

Script file

1. Click Add a file

The screenshot shows the Hyperledger Composer Playground interface. The top navigation bar includes 'Web donation-tracking', 'Define', 'Test', and a user profile 'admin'. The left sidebar contains a 'FILES' section with 'About' (README.md, package.json), 'Model File' (models/model.cto), and 'Access Control' (permissions.ac). The 'Access Control' tab is selected, and the 'Add a file...' button is highlighted with a red box. Below the sidebar, the 'UPDATE NETWORK' section shows a deployment from version 0.0.1 to 0.0.2-deploy.0. The main area displays the 'ACL File permissions.ac' script, which is identical to the one in the first image. A green checkmark and the message 'Everything looks good!' are visible at the bottom of the main area.

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

Add a file

Upload a file from your computer...

Drop here to upload or [browse](#)

☐ **Model File (.cto)**
Define Assets, Participants and Transactions using Hyperledger Composer modelling language.

☒ **Script File (.js)**
Define the logic of transaction executions using JavaScript.

☐ **Query File (.qry)**
Define the queries in here (Note: you can only have 1 of these per .bna).

☐ **Access Control File (permissions.acl)**
Define your access controls here (Note: you can only have 1 of these per .bna).

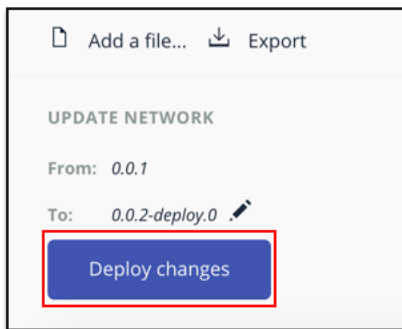
Cancel Add

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.

```
Script File lib/script.js
1 'use strict';
2 /**
3  * Write your transaction processor functions here
4  */
5 var NS = 'org.global.citizens.net';
6 /**
7  * createProjectPledge
8  * @param {org.global.citizens.net.CreateProjectPledge} createProjectPledge
9  * @transaction
10 */
11 function createProjectPledge(txParams) {
12   if(!txParams.name || (txParams.name && txParams.name === "")) {
13     throw new Error('Invalid Pledge Name!!');
14   }
15   if(!txParams.aidOrg) {
16     throw new Error('Invalid Aid Org!!');
17   }
18   var factory = getFactory();
19   var pledge = null;
20   return getAssetRegistry(NS + '.ProjectPledge').then(function (registry) {
```

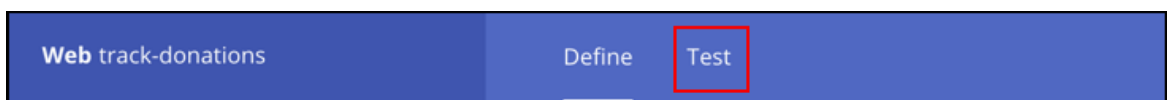
4. After adding all the code snippets to the Model, Script and ACL files, click **Deploy Changes**



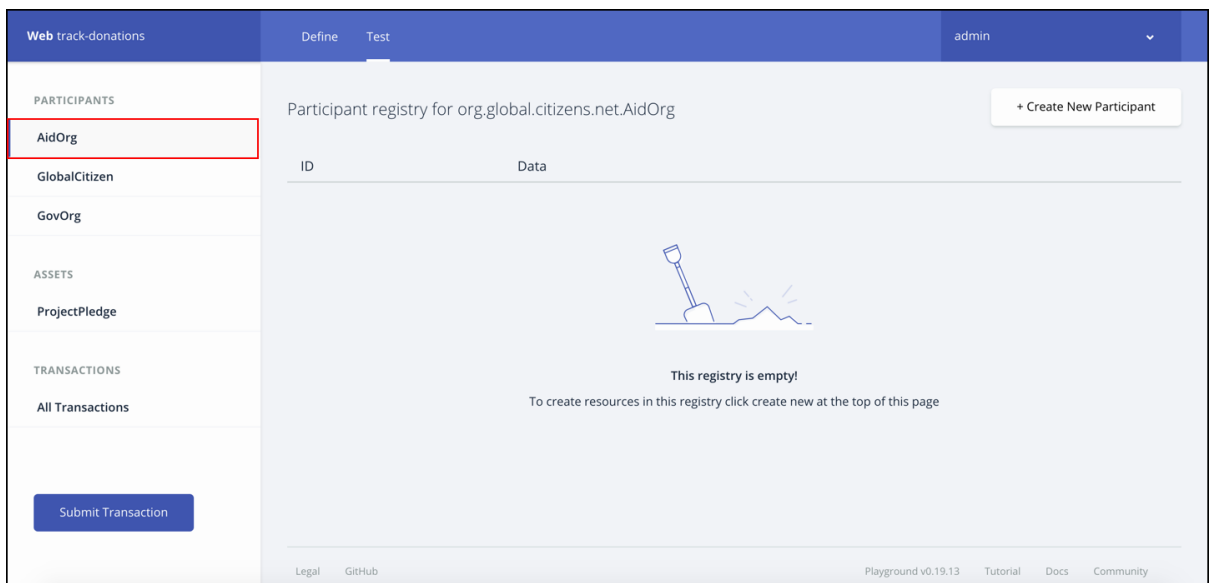
Add Participants

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": []  
}
```

Create New Participant

In registry: **org.global.citizens.net.AidOrg**

JSON Data Preview

```
1 {  
2   "$class": "org.global.citizens.net.AidOrg",  
3   "aidOrgId": "aid",  
4   "projectPledge": []  
5 }
```

☐ Optional Properties

Just need quick test data? [Generate Random Data](#)

Cancel

Create New

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

Web track-donations

Define Test

admin

PARTICIPANTS

AidOrg

GlobalCitizen

GovOrg

ASSETS

ProjectPledge

TRANSACTIONS

All Transactions

Submit Transaction

Participant registry for org.global.citizens.net.GlobalCitizen

+ Create New Participant

ID	Data
<div><p>This registry is empty!</p><p>To create resources in this registry click create new at the top of this page</p></div>	

Legal GitHub

Playground v0.19.13 Tutorial Docs Community

6. Click **Create New Participant**

Participant registry for org.global.citizens.net.GlobalCitizen

+ 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": []
}
```

Create New Participant

In registry: **org.global.citizens.net.GlobalCitizen**

JSON Data Preview

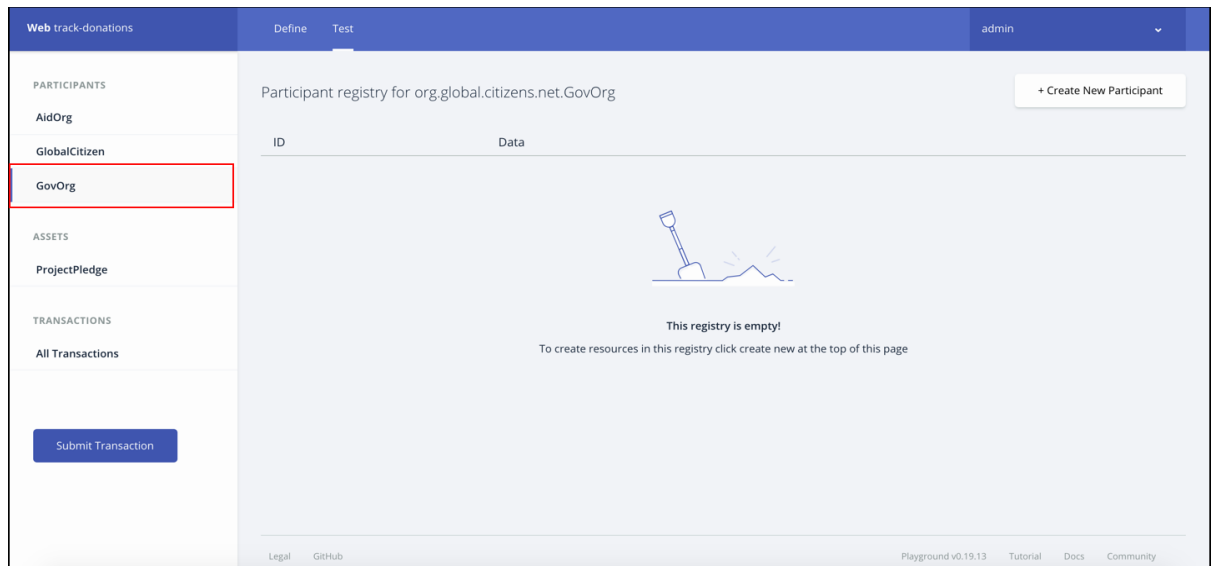
```
1 {
2   "$class": "org.global.citizens.net.GlobalCitizen",
3   "citizenId": "Citizen",
4   "projectPledge": []
5 }
```

☐ Optional Properties

Just need quick test data? [Generate Random Data](#)

Cancel Create New

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": []
}
```

Create New Participant

In registry: **org.global.citizens.net.GovOrg**

JSON Data Preview

```
1 {
2   "$class": "org.global.citizens.net.GovOrg",
3   "govOrgId": "Gov",
4   "fundedPledges": [],
5   "projectPledge": []
6 }
```

☐ Optional Properties

Just need quick test data? [Generate Random Data](#)

Cancel

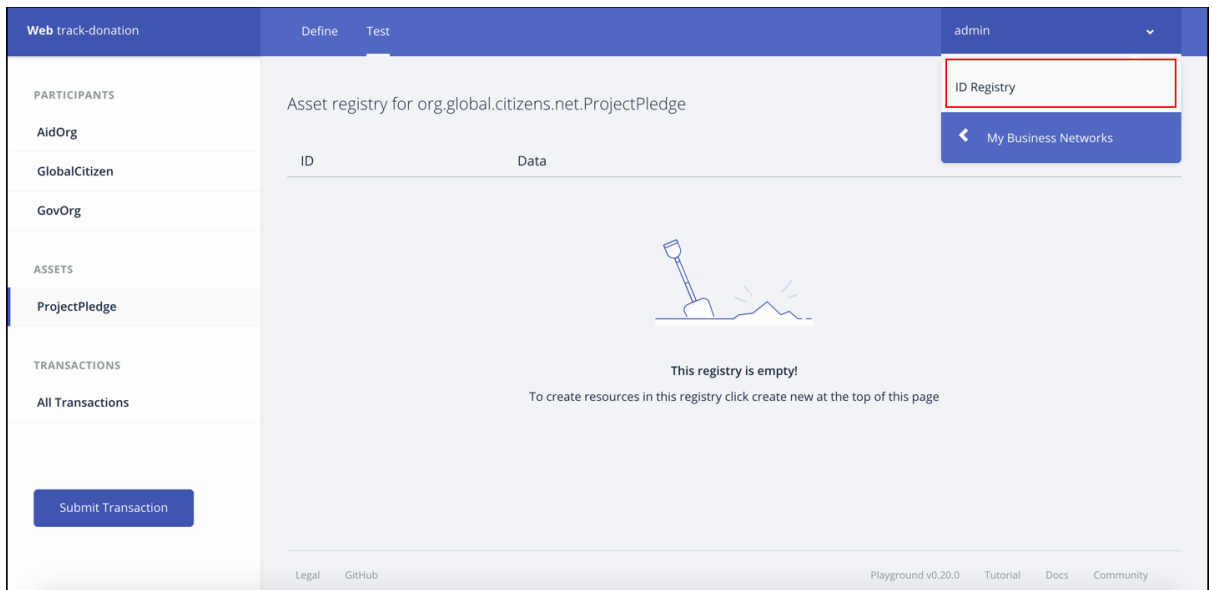
Create New

Add Network Cards

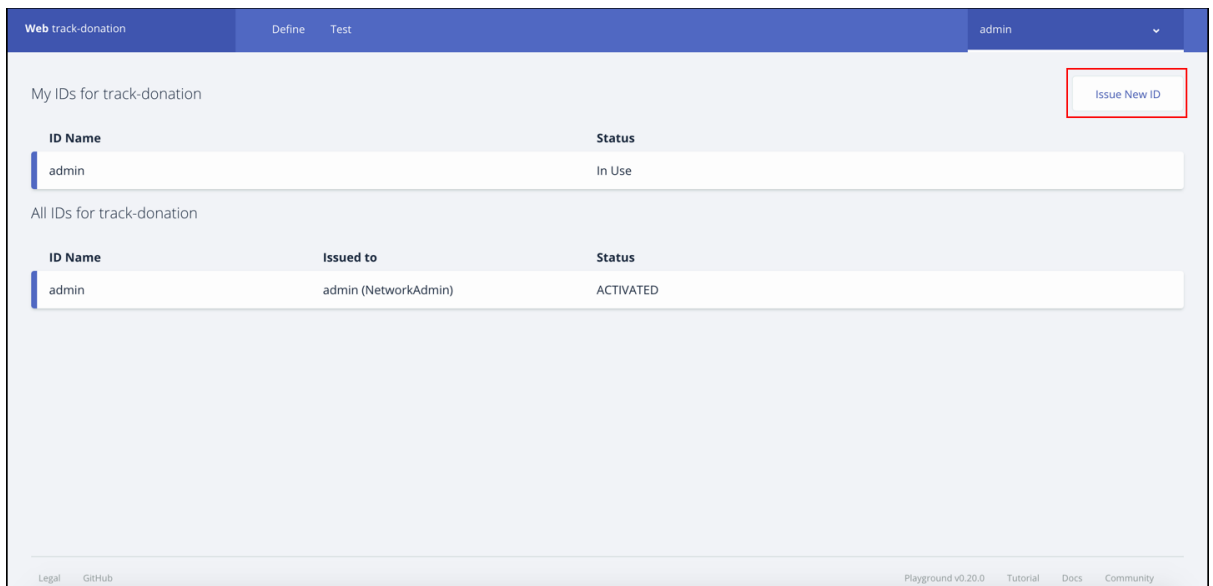
Connection profiles contain the information necessary to connect to a fabric. **Business Network Cards** combine a connection profile, identity, and certificates to allow a connection to a business network in Hyperledger Composer Playground.

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

Issue New Identity

Issue a new ID to a participant in your business network

ID Name*

aid

Participant*

a|

Aid AidOrg

Citizen GlobalCitizen

Gov GovOrg

☐ Allow this

Issuing an identity to a participant who has been issued an identity already or use it yourself when

Cancel

Create New

4. Assign the ID and click create new

Issue New Identity

Issue a new ID to a participant in your business network

ID Name*

Citizen

Participant*

d|

Aid AidOrg

Citizen GlobalCitizen

Gov GovOrg

☐ Allow this

Issuing an identity to a participant who has been issued an identity already or use it yourself when

Cancel

Create New

5. Assign the ID and click create new

Issue New Identity

Issue a new ID to a participant in your business network

ID Name*

Gov

Participant*

gl

Aid AidOrg

Citizen GlobalCitizen

Gov GovOrg

☐ Allow this

Issuing an identity that has been issued

or use it yourself when

Cancel

Create New

- Once the ID's are created you will be able to view them in the ID Registry

Web track-donations

DefineTest

admin

My IDs for track-donations

Issue New ID

ID Name	Status
admin	In Use
aid	In my wallet
Citizen	In my wallet
Gov	In my wallet

My IDs for track-donations

ID Name	Issued to	Status
admin	admin (NetworkAdmin)	ACTIVATED
aid	Aid (AidOrg)	ISSUED
Citizen	Citizen (GlobalCitizen)	ISSUED
Gov	Gov (GovOrg)	ISSUED

Legal

GitHub

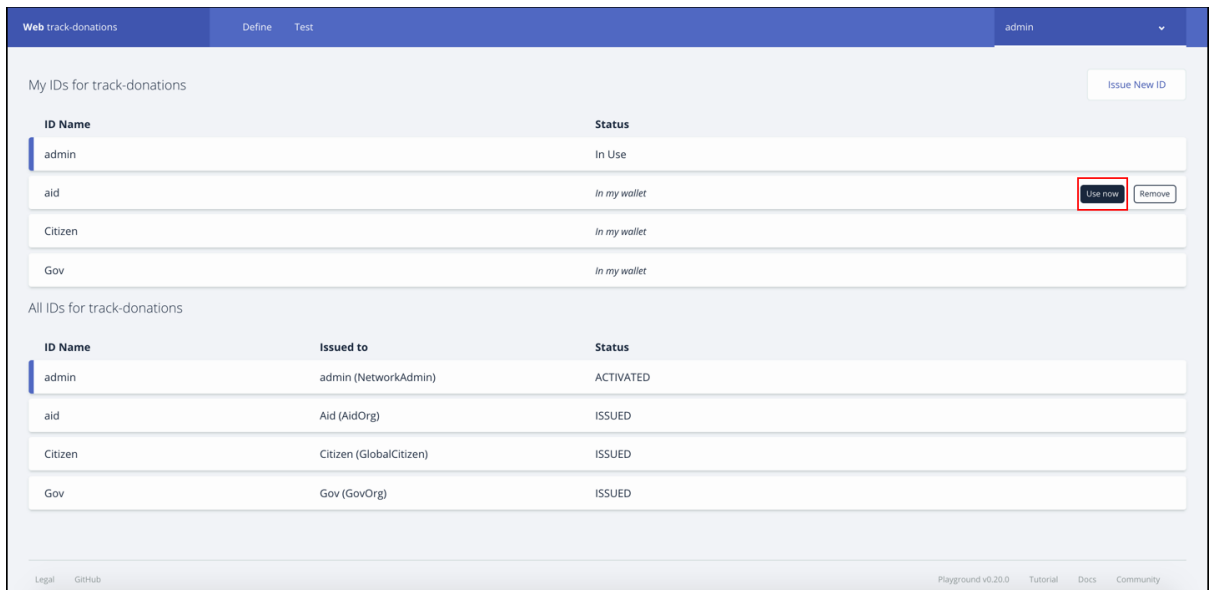
Playground v0.20.0

Tutorial

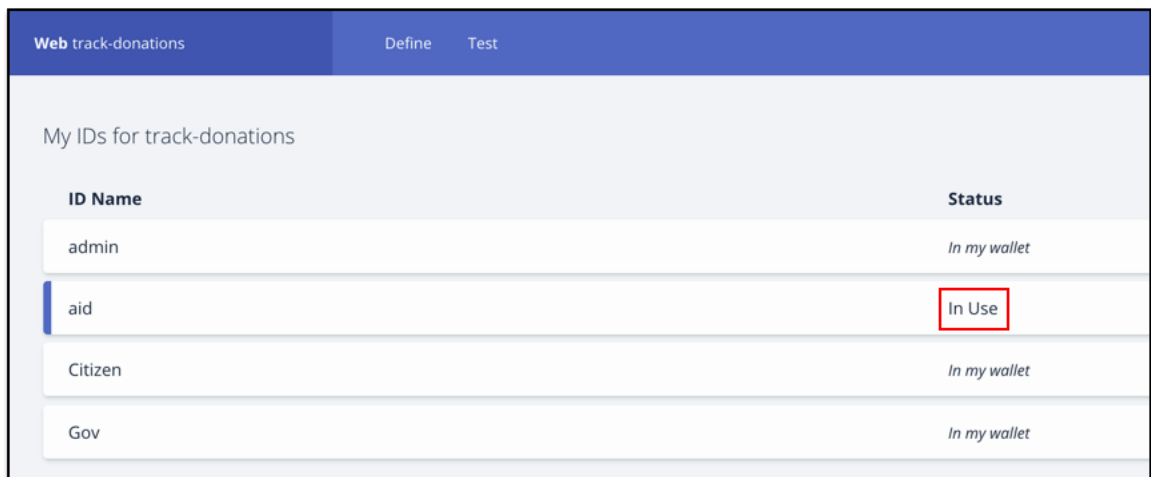
Docs

Community

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



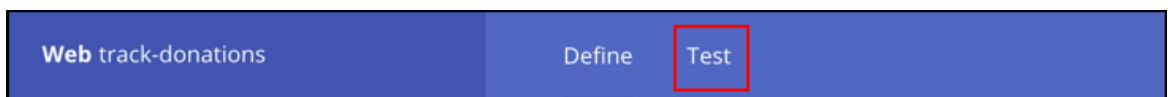
8. You will be able to change in the status once you click on **Use Now**



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.

Web track-donation

PARTICIPANTS

AidOrg

GlobalCitizen

GovOrg

ASSETS

ProjectPledge

TRANSACTIONS

All Transactions

Submit 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"
}
```

Submit Transaction

Transaction Type

CreateProjectPledge

JSON Data Preview

```
1 {
2   "$class": "org.global.citizens.net.CreateProjectPledge",
3   "pledgeId": "p1",
4   "name": "child care",
5   "description": "child care fund",
6   "fundsRequired": 100000,
7   "aidOrg": "resource:org.global.citizens.net.AidOrg#aid"
8 }
```

☐ Optional Properties

Just need quick test data? [Generate Random Data](#)

Cancel

Submit

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

Web track-donation

Define Test

admin

PARTICIPANTS

AidOrg

GlobalCitizen

GovOrg

ASSETS

ProjectPledge

TRANSACTIONS

All Transactions

Submit Transaction

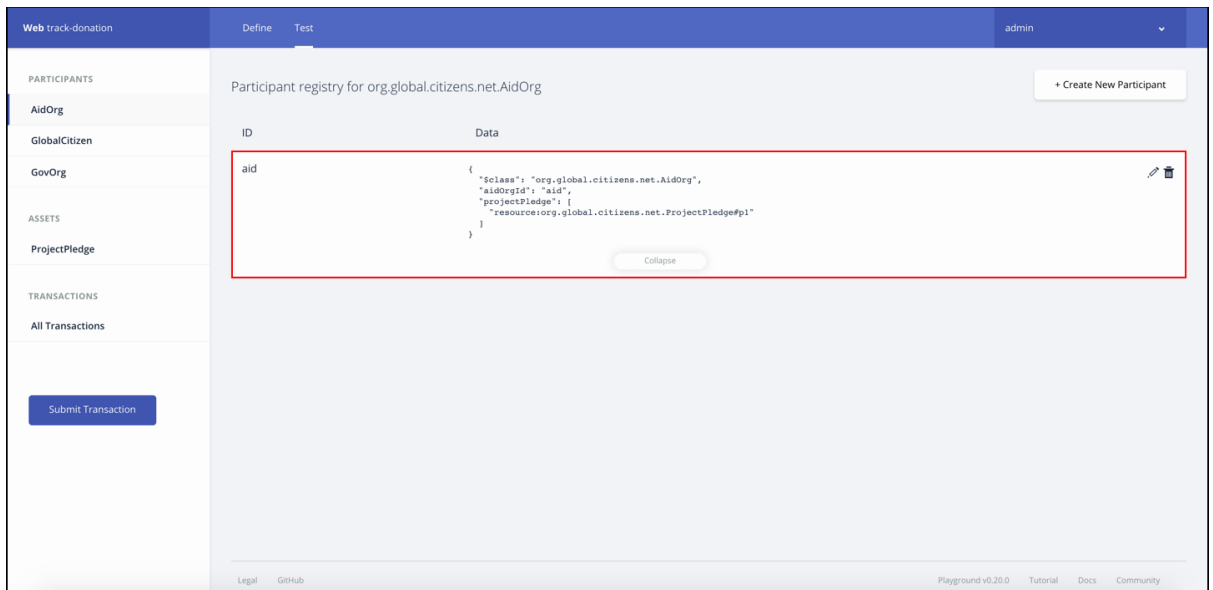
Asset registry for org.global.citizens.net.ProjectPledge

+ Create New Asset

ID	Data
p1	<div><pre>{ "\$class": "org.global.citizens.net.ProjectPledge", "pledgeId": "p1", "name": "child care", "description": "child care fund", "fundsRequired": 100000, "status": "INITIALSTATE", "aidOrg": "resource:org.global.citizens.net.AidOrg#aid", "funds": [] }</pre></div>

Collapse

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"
}
```

Submit Transaction

Transaction Type

SendPledgeToGlobalCitizen

JSON Data Preview

```

1 {
2   "$class": "org.global.citizens.net.SendPledgeToGlobalCitizen",
3   "citizenId":
4   "resource:org.global.citizens.net.GlobalCitizen#Citizen",
5   "pledgeId": "resource:org.global.citizens.net.ProjectPledge#p1"
6 }

```

☐ Optional Properties

Just need quick test data? [Generate Random Data](#)

Cancel

Submit

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

Web track-donation

Define Test

admin

PARTICIPANTS

AidOrg

GlobalCitizen

GovOrg

ASSETS

ProjectPledge

TRANSACTIONS

All Transactions

Submit Transaction

Participant registry for org.global.citizens.net.GlobalCitizen

+ Create New Participant

ID	Data
Citizen	<pre> { "\$class": "org.global.citizens.net.GlobalCitizen", "citizenId": "Citizen", "projectPledge": { "resource:org.global.citizens.net.ProjectPledge#p1" } } </pre>

Collapse

Legal

GitHub

Playground v0.20.0

Tutorial

Docs

Community

- 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"
}
```

Submit Transaction

Transaction Type: **SendPledgeToGovOrg**

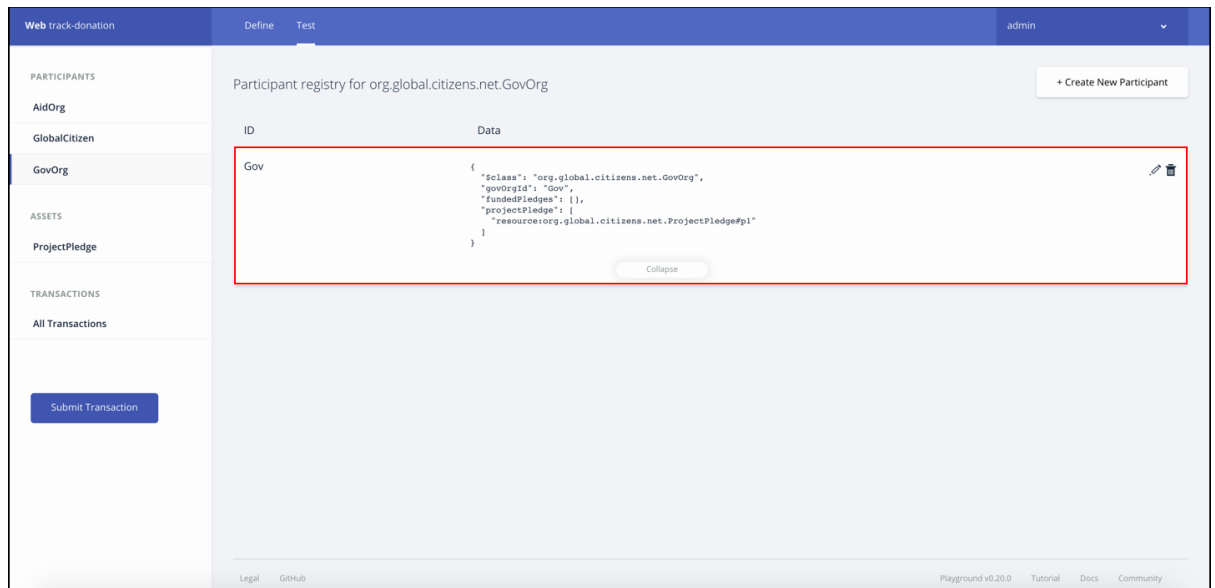
JSON Data Preview

```
1 {
2   "$class": "org.global.citizens.net.SendPledgeToGovOrg",
3   "govOrg": ["resource:org.global.citizens.net.GovOrg#gov"],
4   "pledgeId": "resource:org.global.citizens.net.ProjectPledge#p1"
5 }
```

☐ Optional Properties

Just need quick test data? [Generate Random Data](#) Cancel **Submit**

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
}
```

Submit Transaction

Transaction Type UpdatePledge

JSON Data Preview

```
1 {
2   "$class": "org.global.citizens.net.UpdatePledge",
3   "govOrgId": "resource:org.global.citizens.net.GovOrg#Gov",
4   "pledgeId": "resource:org.global.citizens.net.ProjectPledge#p1",
5   "fundingType": "WEEKLY",
6   "approvedFunding": 100000,
7   "fundsPerInstallment": 1000
8 }
```

☐ Optional Properties

Just need quick test data? [Generate Random Data](#)

Cancel

Submit

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

Web track-donation

Define Test

admin

PARTICIPANTS

AidOrg

GlobalCitizen

GovOrg

ASSETS

ProjectPledge

TRANSACTIONS

All Transactions

Submit Transaction

Participant registry for org.global.citizens.net.GovOrg

+ Create New Participant

ID	Data
Gov	<pre>{ "\$class": "org.global.citizens.net.GovOrg", "govOrgId": "Gov", "fundedPledges": ["resource:org.global.citizens.net.ProjectPledge#p1"], "projectPledge": { "resource:org.global.citizens.net.ProjectPledge#p1" } }</pre>

Collapse

Legal

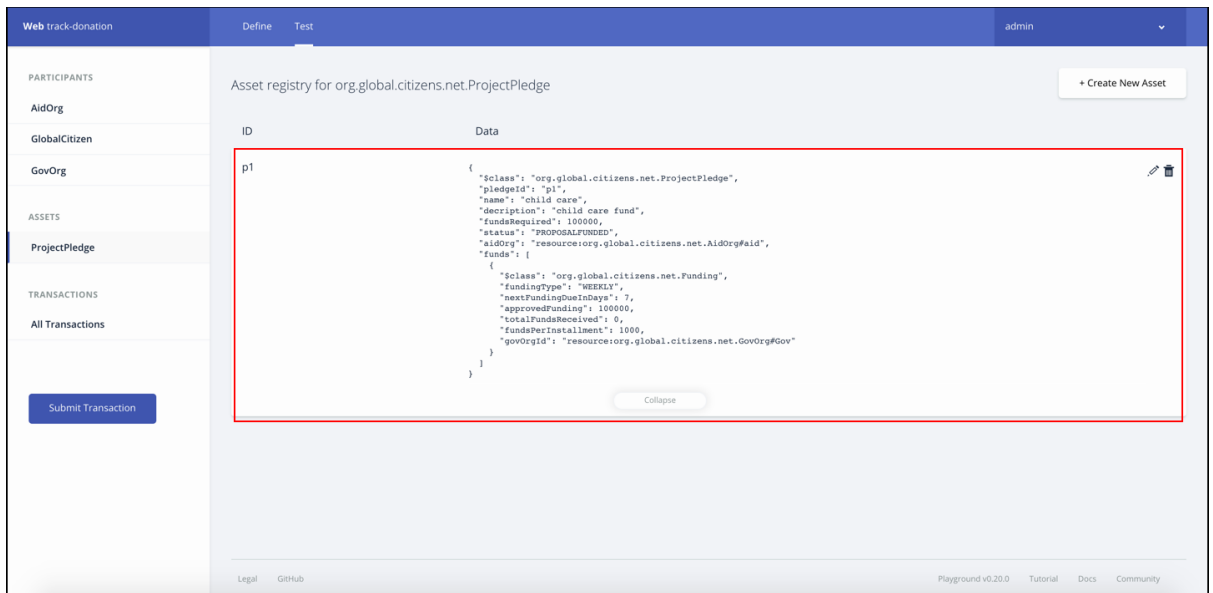
GitHub

Playground v0.20.0

Tutorial

Docs

Community



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"
}
```

Submit Transaction

Transaction Type

TransferFunds

JSON Data Preview

```
1 {
2   "$class": "org.global.citizens.net.TransferFunds",
3   "govOrgId": "resource:org.global.citizens.net.GovOrg#Gov",
4   "pledgeId": "resource:org.global.citizens.net.ProjectPledge#p1"
5 }
```

☐ Optional Properties

Just need quick test data? [Generate Random Data](#)

Cancel

Submit

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

Web track-donation

Define Test

admin

PARTICIPANTS

AidOrg

GlobalCitizen

GovOrg

ASSETS

ProjectPledge

TRANSACTIONS

All Transactions

Submit Transaction

Asset registry for org.global.citizens.net.ProjectPledge

+ Create New Asset

ID	Data
p1	<div><pre>{ "\$class": "org.global.citizens.net.ProjectPledge", "pledgeId": "p1", "name": "child care", "description": "child care fund", "fundsRequired": 100000, "status": "PROPOSALFORWARDED", "aidOrg": "resource:org.global.citizens.net.AidOrg#aid", "funds": [{ "\$class": "org.global.citizens.net.Funding", "fundingType": "WEEKLY", "nextFundingDueInDays": 7, "approvedFunding": 100000, "totalFundsReceived": 1000, "fundsPerInstallment": 1000, "govOrgId": "resource:org.global.citizens.net.GovOrg#Gov" }] }</pre></div> <div>Collapse</div>

Legal

GitHub

Playground v0.20.0

Tutorial

Docs

Community