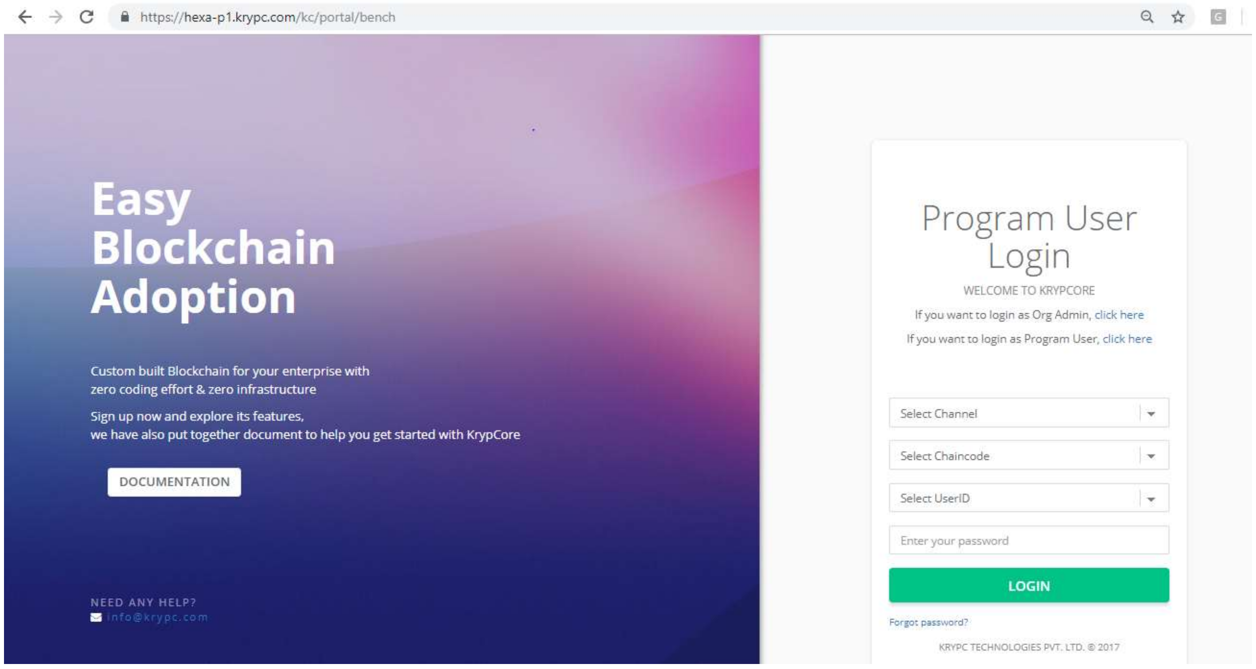
KRYPCORE USER MANUAL

Contents

MAPPING USE CASE/PROGRAM DEFINITION

Define use case / Program name:

After having logged in as the Bench administrator, the first step is to define / configure use case name



https://hexa-p1.krypc.com/kc/portal/bench

Easy Blockchain Adoption

Custom built Blockchain for your enterprise with zero coding effort & zero infrastructure

Sign up now and explore its features, we have also put together document to help you get started with KrypCore

DOCUMENTATION

NEED ANY HELP? info@krypc.com

Program User Login

WELCOME TO KRYPCORE

If you want to login as Org Admin, click here

If you want to login as Program User, click here

Select Channel

Select Chaincode

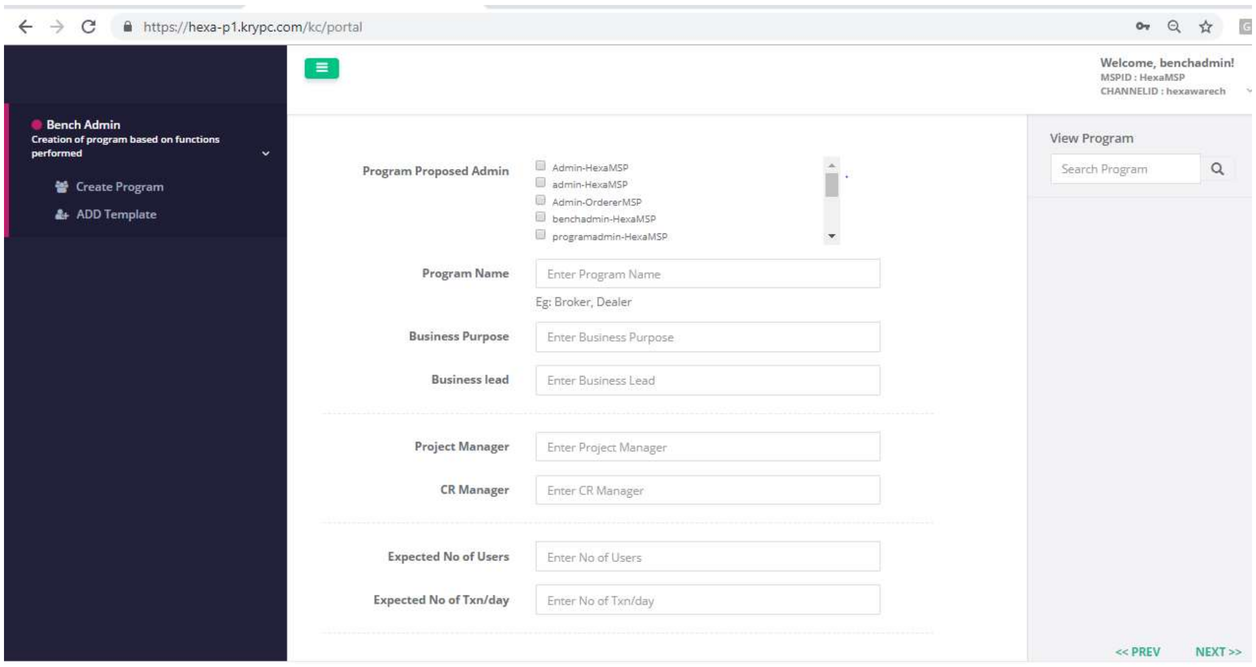
Select UserID

Enter your password

LOGIN

Forgot password?

KRYPC TECHNOLOGIES PVT. LTD. © 2017



https://hexa-p1.krypc.com/kc/portal

Welcome, benchadmin!
MSPID : HexaMSP
CHANNELID : hexawarech

Bench Admin
Creation of program based on functions performed

Create Program

ADD Template

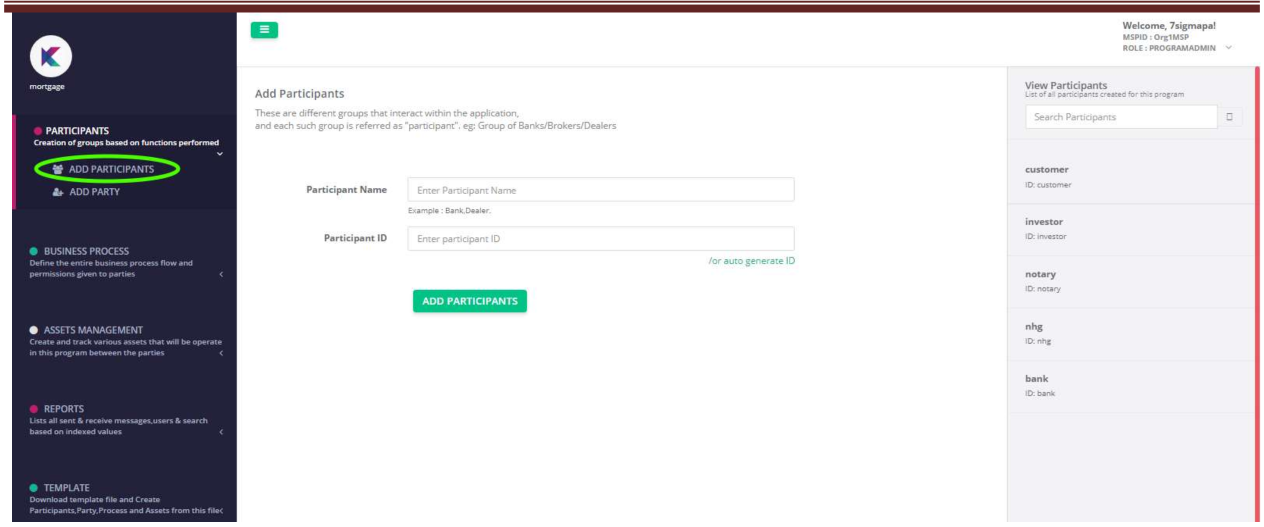
Program Proposed Admin	Admin-HexaMSP admin-HexaMSP Admin-OrdererMSP benchadmin-HexaMSP programadmin-HexaMSP
Program Name	Enter Program Name Eg: Broker, Dealer
Business Purpose	Enter Business Purpose
Business lead	Enter Business Lead
Project Manager	Enter Project Manager
CR Manager	Enter CR Manager
Expected No of Users	Enter No of Users
Expected No of Txn/day	Enter No of Txn/day

View Program

Search Program

<< PREV NEXT >>

Step 1: ADD PARTICIPANTS:


mortgage
PARTICIPANTS
Creation of groups based on functions performed
ADD PARTICIPANTS
ADD PARTY
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORTS
Lists all sent & receive messages,users & search based on indexed values
TEMPLATE
Download template file and Create Participants,Party,Process and Assets from this filed
Welcome, 7sigmapa!
MSPID : Org1MSP
ROLE : PROGRAMADMIN
ADD PARTICIPANTS
View Participants
Search Participants

Add Participants

These are different groups that interact within the application,
and each such group is referred as "participant". eg: Group of Banks/Brokers/Dealers

Participant Name	Enter Participant Name
	Example : Bank,Dealer.
Participant ID	Enter participant ID
	/or auto generate ID

List of all participants created for this program

customer
ID: customer

investor
ID: investor

notary
ID: notary

nhg
ID: nhg

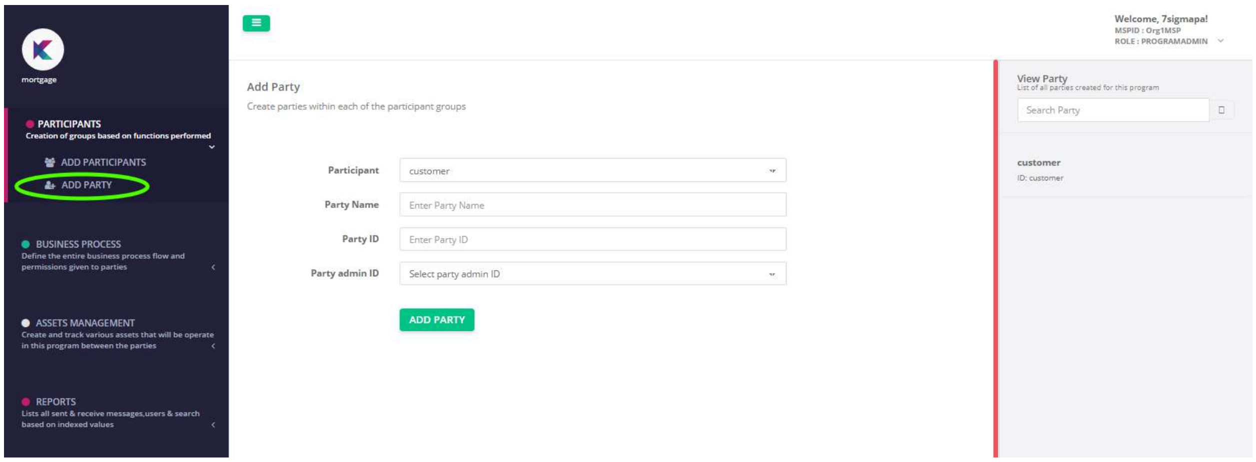
bank
ID: bank

After having logged in as the program administrator, the first step is to create the identified participants of the chosen use case. In KrypCore application, a "***participant***" is referred to a superset differentiated based on the core functional areas that uniquely defines a group in the use case. These groups principally carry out or have execution rights to a set of processes in the use case. Examples of group in a Mortgage use case can be banks, agencies, sellers, buyers etc or in a typical supply chain use case the participants can be the supplier, receiver, logistics, banker, regulator etc.

To add participants, click on "ADD PARTICIPANTS" tab on the functional menu-bar followed by:

- Enter Participant Name (Name the identified functional group in the use-case)
- Enter Participant ID (Give a unique reference to the identified group)

Step 2: ADD PARTY


mortgage
PARTICIPANTS
Creation of groups based on functions performed
ADD PARTICIPANTS
ADD PARTY
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORTS
Lists all sent & receive messages,users & search based on indexed values
Welcome, 7sigmapa!
MSPID : Org1MSP
ROLE : PROGRAMADMIN
ADD PARTY
View Party
Search Party

Add Party

Create parties within each of the participant groups

Participant	customer
Party Name	Enter Party Name
Party ID	Enter Party ID
Party admin ID	Select party admin ID

List of all parties created for this program

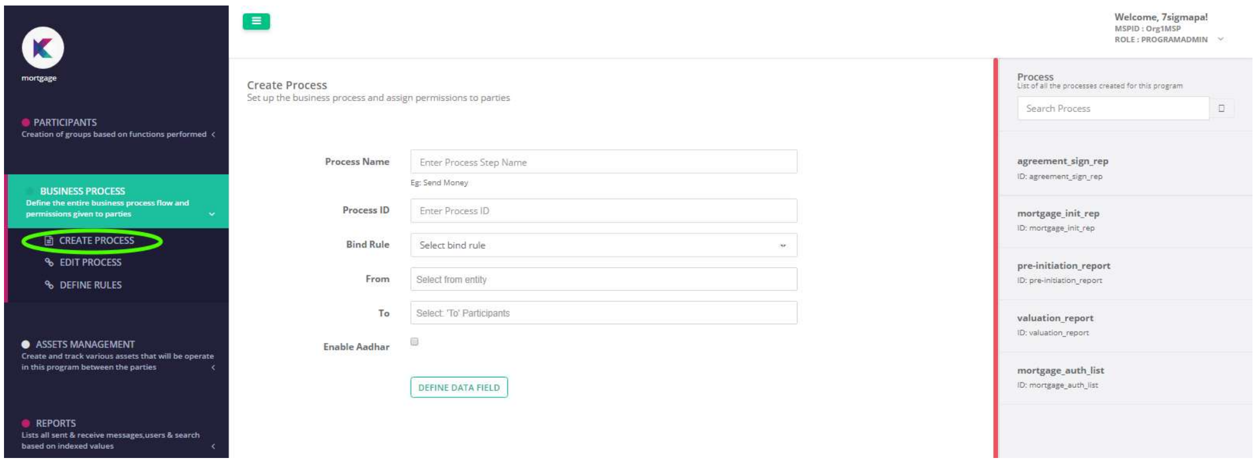
customer
ID: customer

Next step to carry out is to add "**Party/ies**". A "**party**" in KrypCore application typically represents a legal entity or an organization joining or has affirmed to join the use-case eco-system. These legal entities are placed within a functional group or "**participant**" based on the functionalities they are supposed to carry out within the use-case eco-system. Thus, all "**parties**" in a specific "**participant**" group automatically inherit the process step or event specific execution rights provided to that "**participant**".

To add parties, click on the “**ADD PARTY**” tab on the functional menu-bar followed by entry of:
Select the participant group under which you want to create the party

- Provide a party Name
- Input a unique Party ID that will refer to the Party/ Entity
- Map the Party Admin ID from the list of user that is available in the wallet manager. Ensure the user mapped can only act the party admin within the framework of the use case , you cannot allocate the same user an end user permission

STEP 3: CREATE PROCESS


mortgage
PARTICIPANTS
Creation of groups based on functions performed
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
CREATE PROCESS
EDIT PROCESS
DEFINE RULES
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORTS
Lists all sent & receive messages,users & search based on indexed values
Welcome, 7sigmapa!
MSPID : Org1MSP
ROLE : PROGRAMADMIN
DEFINE DATA FIELD

Create Process
Set up the business process and assign permissions to parties

Process Name	Enter Process Step Name Eg: Send Money
Process ID	Enter Process ID
Bind Rule	Select bind rule
From	Select from entity
To	Select 'To' Participants
Enable Aadhar	☐

Process
List of all the processes created for this program
Search Process

agreement_sign_rep
ID: agreement_sign_rep

mortgage_init_rep
ID: mortgage_init_rep

pre-initiation_report
ID: pre-initiation_report

valuation_report
ID: valuation_report

mortgage_auth_list
ID: mortgage_auth_list

Create Process is a functionality provided to map in the identified processes or events that needs to be captured during the course of running the program. Events or Processes can only be mapped after the addition of participants. These processes can be classified broadly in two types:

- **Invoke Events/Process**: These types of transactions are the ones which allow writing data into the inherent blockchain network

- **Report Events/Process**: These types of transactions are basically query transactions that allow querying the data lake synced to the transactions recorded in blockchain network against set of rules or query parameters. Executing these types of transactions does not add/record data into the blockchain network.

To add events or processes, click on the "**CREATE PROCESS**" tab within the **BUSINESS PROCESS** group. The inputs required to create a process will include:

- **Process Name**: A name representing the process can be entered here
- **Process ID**: Enter a unique reference to the process that is getting mapped. This unique reference is further used while coding in the business rules and uniquely identifies a process step in the program
- **Bind Rule**: You can select on the multiple bind rules provided, a brief on each of the rule is appended below:
  - ➢ REPORT_WALLET: (Report)
    Tells the application that the event getting defined is a query event thus does not require any recording of data into the blockchain network.

  - ➢ REPORT_LOCAL: (Report)

  - ➢ Spend referred message: (Invoke)
    This rule can be selected in case the current event execution is based execution of a past event and can only be executed once against the referred event.

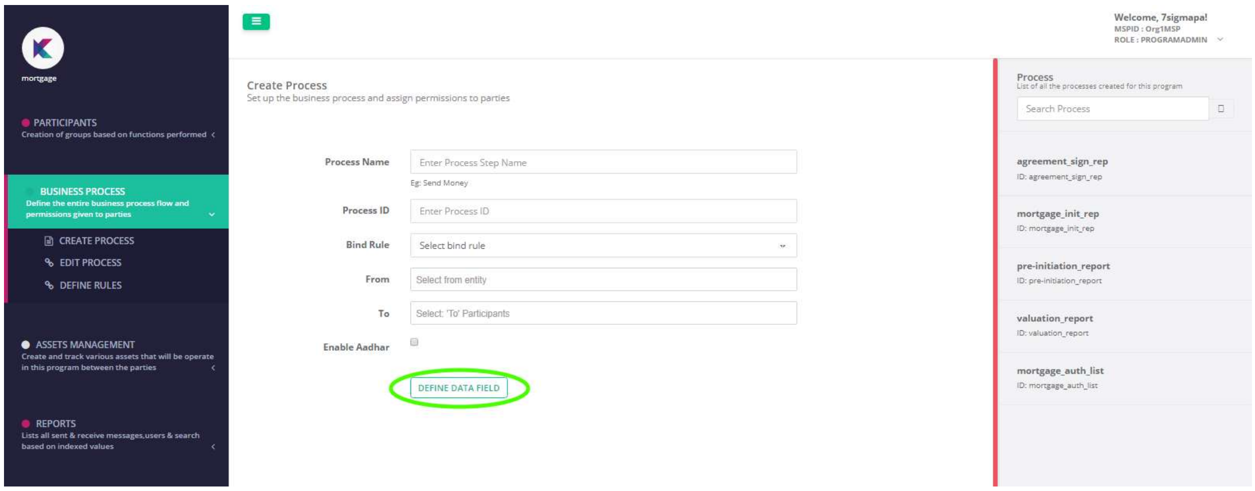
  - ➢ None: (Invoke)

This rule is selected when the event is defined as an independent process which need not refer to any previous event before getting executed.

- ➢ Spend referred msg and reply to sender only:
  Rule applied when an event requires the user to perform an action by tagging it to a reference event, the reference event can be executed only once and action taken is intimated only to the originator.

- ➢ Refer a message: (Invoke)
  Rule applied on an event/process when there is a need to refer a previous event that has occurred and link it to the current action

- **From**: Allows the program admin to specify the execution rights of this event/process to participant/s
- **To**: The recipient participant/s can be selected from the list of the existing participants. This allows specifying the intended recipient participants of the event.


mortgage
PARTICIPANTS
Creation of groups based on functions performed
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
CREATE PROCESS
EDIT PROCESS
DEFINE RULES
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORTS
Lists all sent & receive messages,users & search based on indexed values
Welcome, 7sigmapa!
MSPID : Org1MSP
ROLE : PROGRAMADMIN
DEFINE DATA FIELD
Search Process

Create Process
Set up the business process and assign permissions to parties

Process Name	Enter Process Step Name (Eg: Send Money)
Process ID	Enter Process ID
Bind Rule	Select bind rule
From	Select from entity
To	Select: 'To' Participants
Enable Aadhar	☐

Process
List of all the processes created for this program

agreement_sign_rep
ID: agreement_sign_rep

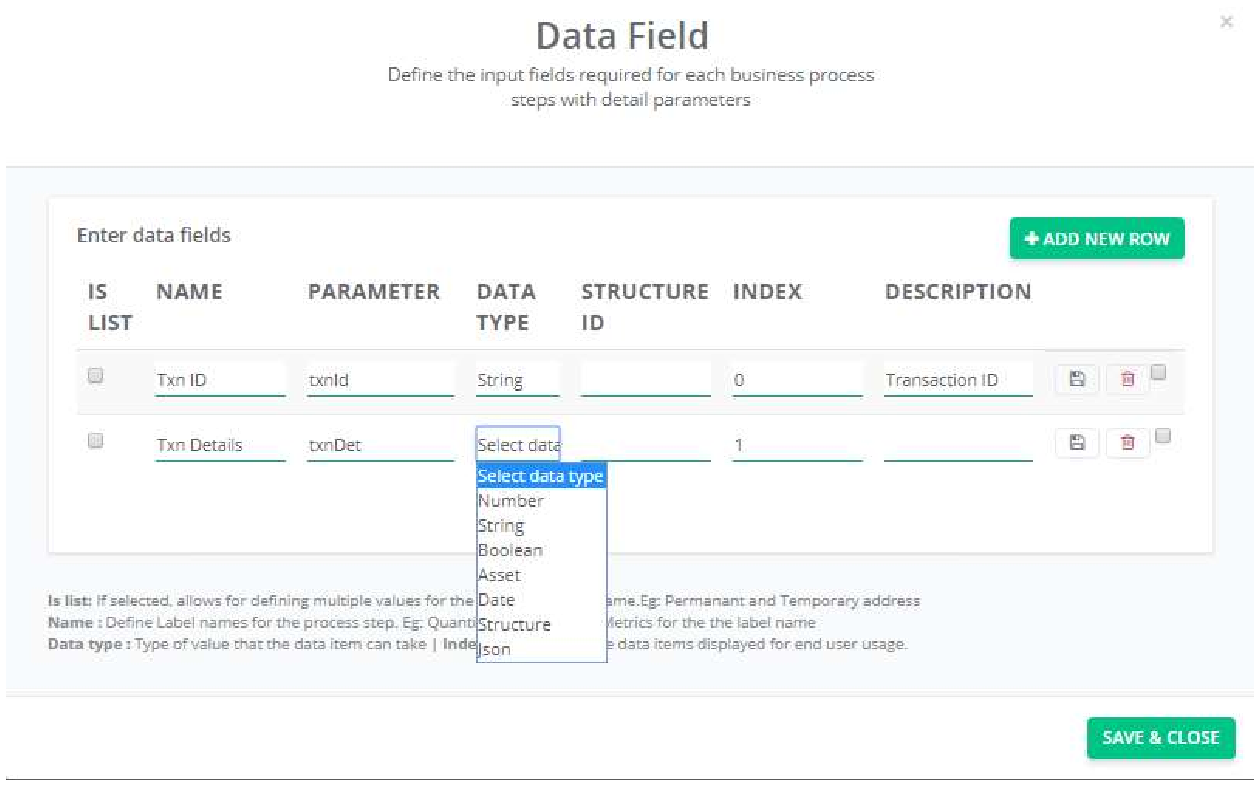
mortgage_init_rep
ID: mortgage_init_rep

pre-initiation_report
ID: pre-initiation_report

valuation_report
ID: valuation_report

mortgage_auth_list
ID: mortgage_auth_list

- Next step in Creation/mapping of a process is defining the parameters that form the payload of the process. You can proceed to defining the parameters of a specific event by clicking on the "**DEFINE DATA FIELD**" tab



Data Field

Define the input fields required for each business process steps with detail parameters

Enter data fields

IS LIST	NAME	PARAMETER	DATA TYPE	STRUCTURE ID	INDEX	DESCRIPTION
	Txn ID	txnId	String		0	Transaction ID
	Txn Details	txnDet	Select data		1	

Is list: If selected, allows for defining multiple values for the ame.Eg: Permanant and Temporary address
Name : Define Label names for the process step. Eg: Quanti letrics for the the label name
Data type : Type of value that the data item can take | **Inde** e data items displayed for end user usage.

Input screen to define data parameters

The elements available to define data fields are appended below:
ISLIST: Allows the designer to specify if the parameters will contain a list of values. For example, if during execution of an event a parameter needs to capture three strings together in a list like [red,green,amber], one needs to enable the IS LIST button against the parameter and specify the DATA TYPE as String

NAME: This field will contain the name of the field

PARAMETER: It forms a unique reference to the named field and will be used as a reference during coding in the business rules

DATA TYPE: It provides for the type of data that this field will take in during the execution of the event. The list of data types currently available are:

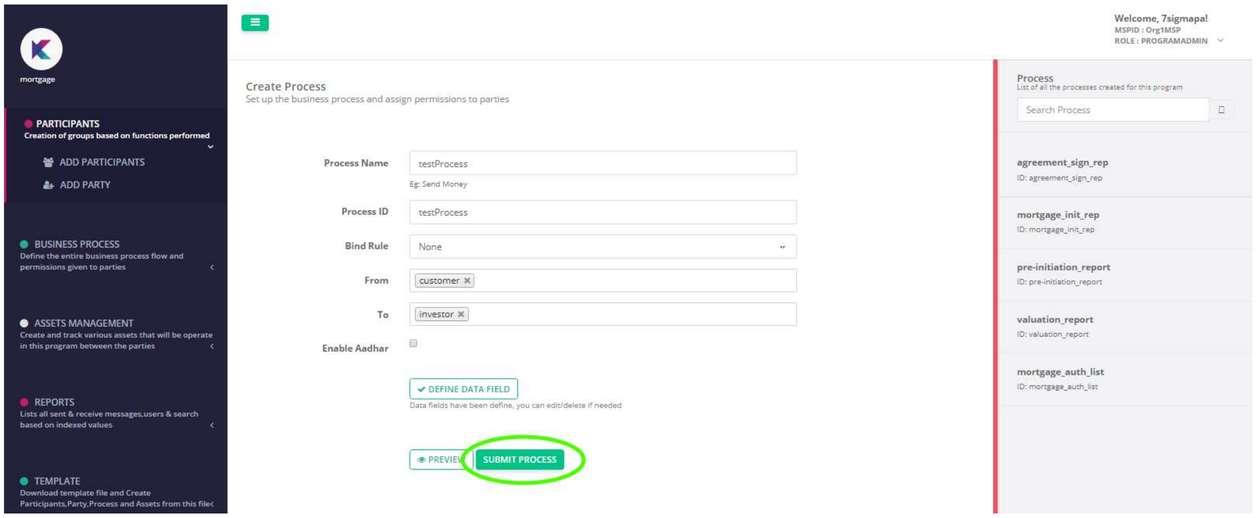
Type	**Max Length**
Number	🟨
String	🟨
Boolean	🟨
Asset	🟨
Date	🟨
Structure	🟨
JSON	🟨

INDEX: The value of Index gets automatically configured while committing the event in blockchain network. It role is to determine the placement of the data fields in the auto generated executable forms that is later discussed in the document

DESCRIPTION: It's a placeholder which is used to provide a brief description on the respective data field

BIN ICON: This radio button may be used to activate/deactivate a data field. Once deactivated null value automatically gets transmitted during the execution of the event. Also its important to note that a deactivated data field cannot be used or referred in the business logic codes created for this event.

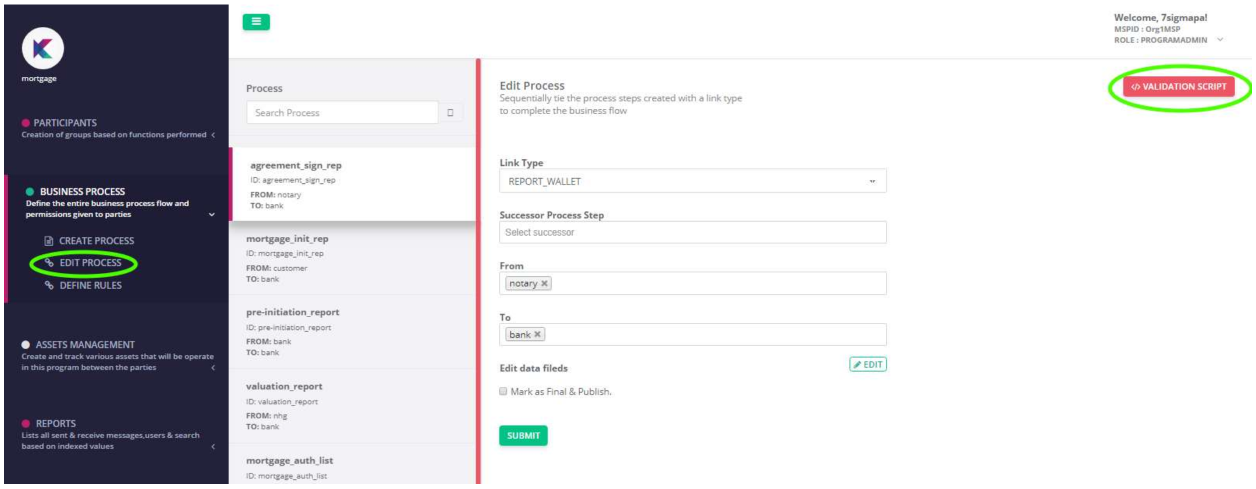
After entering the requisite data fields, click on the **SAVE & CLOSE** button to temporarily save the defined structure.


mortgage
PARTICIPANTS
Creation of groups based on functions performed
ADD PARTICIPANTS
ADD PARTY
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORTS
Lists all sent & receive messages,users & search based on indexed values
TEMPLATE
Download template file and Create Participants,Party,Process and Assets from this file
Welcome, 7sigmapa!
MSPID : Org1MSP
ROLE : PROGRAMADMIN
Create Process
Set up the business process and assign permissions to parties
DEFINE DATA FIELD
Data fields have been define, you can edit/delete if needed
PREVIEW
SUBMIT PROCESS
Process
List of all the processes created for this program
Search Process
agreement_sign_rep
ID: agreement_sign_rep
mortgage_init_rep
ID: mortgage_init_rep
pre-initiation_report
ID: pre-initiation_report
valuation_report
ID: valuation_report
mortgage_auth_list
ID: mortgage_auth_list

Process Name	testProcess
	Eg: Send Money
Process ID	testProcess
Bind Rule	None
From	customer
To	investor
Enable Aadhar	☐

- Commit the defined process by clicking on the "**SUBMIT PROCESS**" button. On successful submission you will get a alert pop up with the blockchain transaction ID of the transaction.

STEP 4: SCRIPTING USE CASE'S BUSINESS RULES


mortgage
PARTICIPANTS
Creation of groups based on functions performed
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
CREATE PROCESS
EDIT PROCESS
DEFINE RULES
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORTS
Lists all sent & receive messages,users & search based on indexed values
Welcome, 7sigmapa!
MSPID : Org1MSP
ROLE : PROGRAMADMIN
VALIDATION SCRIPT
EDIT
SUBMIT

Process

Search Process

agreement_sign_rep
ID: agreement_sign_rep
FROM: notary
TO: bank

mortgage_init_rep
ID: mortgage_init_rep
FROM: customer
TO: bank

pre-initiation_report
ID: pre-initiation_report
FROM: bank
TO: bank

valuation_report
ID: valuation_report
FROM: nhg
TO: bank

mortgage_auth_list
ID: mortgage_auth_list

Edit Process
Sequentially tie the process steps created with a link type to complete the business flow

Link Type: REPORT_WALLET

Successor Process Step: Select successor

From: notary ✕

To: bank ✕

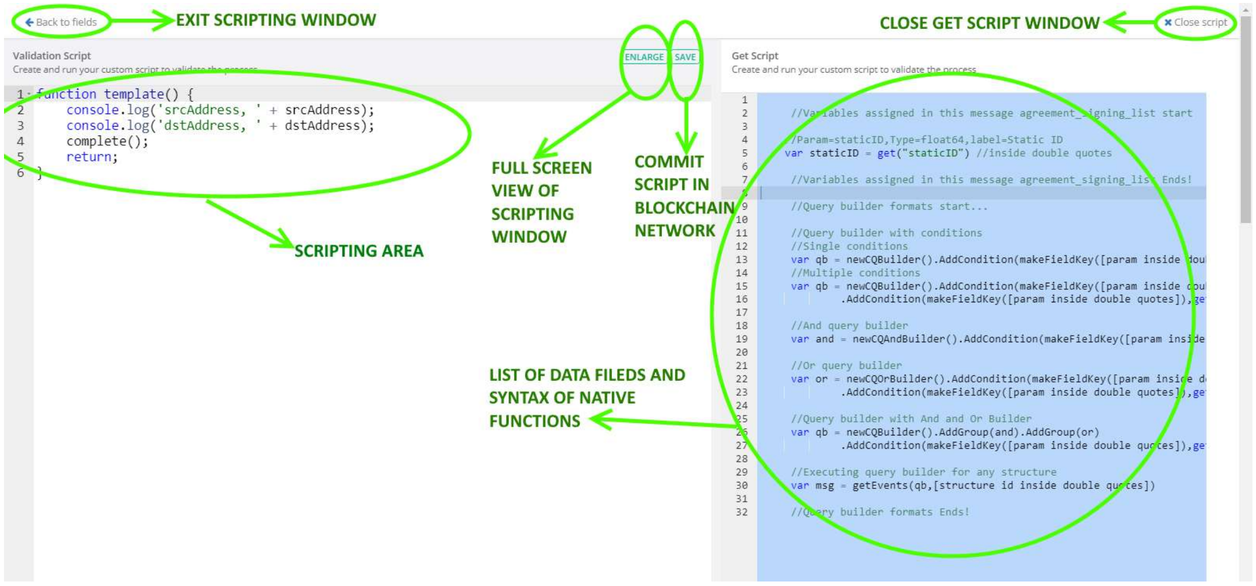
Edit data fileds

☐ Mark as Final & Publish.

- To define use case specific business rules click on "**EDIT PROCESS**" on the functional menu-bar and select the event/process for which specific business rules needs to be scripted

  Other than allowing adding scripts using this function you can also edit the following:
  - Add/remove participants in the "**From**" and "**To**" placeholders
  - Add/Activate/Deactivate data fields of the respective process/event

- Click on "**VALIDATION SCRIPT**" button to get into the scripting window


Back to fields
EXIT SCRIPTING WINDOW
CLOSE GET SCRIPT WINDOW
Close script
ENLARGE
SAVE
FULL SCREEN VIEW OF SCRIPTING WINDOW
COMMIT SCRIPT IN BLOCKCHAIN NETWORK
SCRIPTING AREA
LIST OF DATA FILEDS AND SYNTAX OF NATIVE FUNCTIONS

Validation Script
Create and run your custom script to validate the process

```
function template() {
    console.log('srcAddress, ' + srcAddress);
    console.log('dstAddress, ' + dstAddress);
    complete();
    return;
}
```

Get Script
Create and run your custom script to validate the process

```

    //Variables assigned in this message agreement_signing_list start

   //Param=staticID,Type=float64,label=Static ID
   var staticID = get("staticID") //inside double quotes

    //Variables assigned in this message agreement_signing_list Ends!

    //Query builder formats start...

    //Query builder with conditions
    //Single conditions
    var qb = newCQBuilder().AddCondition(makeFieldKey([param inside dou
    //Multiple conditions
    var qb = newCQBuilder().AddCondition(makeFieldKey([param inside dou
            .AddCondition(makeFieldKey([param inside double quotes]),ge

    //And query builder
    var and = newCQAndBuilder().AddCondition(makeFieldKey([param inside

    //Or query builder
    var or = newCQOrBuilder().AddCondition(makeFieldKey([param inside d
            .AddCondition(makeFieldKey([param inside double quotes]),ge

    //Query builder with And and Or Builder
    var qb = newCQBuilder().AddGroup(and).AddGroup(or)
            .AddCondition(makeFieldKey([param inside double quotes]),ge

    //Executing query builder for any structure
    var msg = getEvents(qb,[structure id inside double quotes])

    //Query builder formats Ends!
```

Native Functions:
KrypCore uses Java scripts to code in specific business rules that need to be adhered to while execution of the respective event/process.

To aid users as well as limit the scope of scripting to business rules only asset of native functionalities are made available. A list of methods along with a brief on each of these functionalities are given in the table below:

S.NO	METHOD	Description
1	makeInteger	returns field value with integer data type
2	makeString	returns field value with string data typ
3	getUuid	returns transaction id
4	setL	sets key value pair for message id
5	getL	returns value for the key for message id
6	setLByMessageId	sets key value pair for specified message id
7	getLByMessageId	returns value for the key of specified message id
8	srcAddress	address of sender of the message
9	dstAddress	address of receiver of the message
10	complete	specifies completion of javascript (not to be used in query mode)
11	setError	to return error from javascript
12	setResult	to return values from javascript
13	get	returns current value of the param of the message
14	getDate	returns current value of param of data type date of the message
15	getUser	returns details of the User
16	newCQBuilder	returns interface for querying data from the state db
17	newCQOrBuilder	returns interface for implementing or condition with the query interface
18	newCQAndBuilder	returns interface for implementing and condition with the query inteface
19	AddGroup	adds `and` or `or` condition interface to the query interface
20	AddCondition	adds condition for search in the query interface
21	getEvents	returns all the events based on query conditions for a specified message Id
21	makeAsset	returns field value with asset data type
22	addDate	adds number of days to a Date
23	compareDate	compares two provided dates(returns 0 if same, -1 if after, 1 if before)
24	toString	returns user identity as pipe(||) separated string
25	compareUser	compares two user identity(returns true if equal)
26	getAsset	returns provided users asset balance
27	transferAsset	transfers an asset from invoker user to specified user
28	executeQuery	returns result based on the created query
29	makeFieldKey	returns state format key for querying purposes
30	makeMessageFieldDateKey	returns state format for date key

31	makeMessageFieldAssetKey	returns state format for asset key
32	callAnotherChaincodeFunction	used to call another chaincode function from the event

OTHER FUNCTIONALITIES:

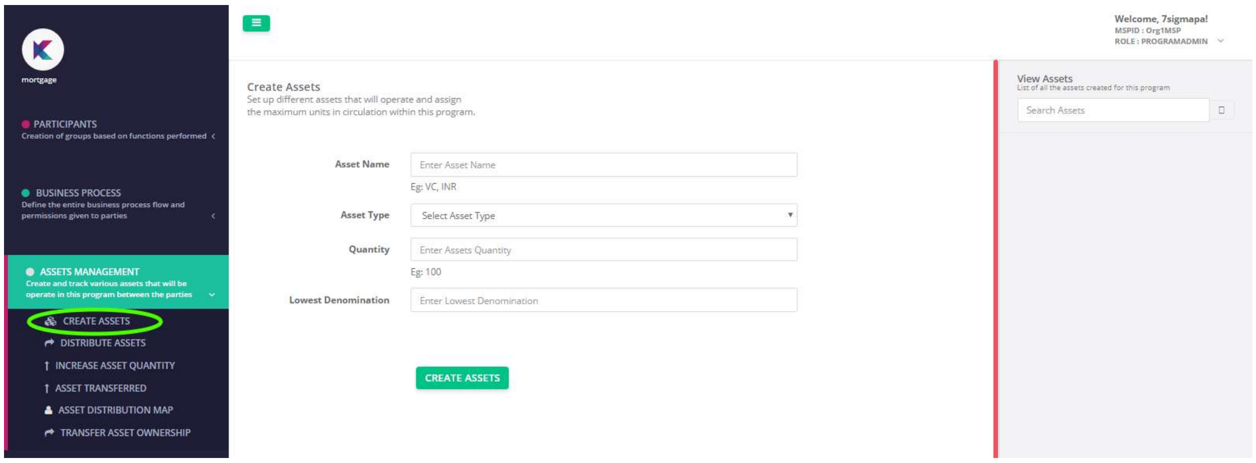
In addition to creation of use case mapping functionalities a list of other functionalities are also provided by Krypcore. These functionalities are described below:

ASSET MANAGEMENT:

An "**asset**" in KrypCore can be defined as an entity that has intrinsic value. The asset thus can represent a digital currency, a painting, a vehicle, points etc. After an asset gets created in KrypCore it automatically gets managed as per general asset management rules which include:

- Management of the ownership of the asset
- Avoidance of double spends
- Availability of these assets as a data type that can be further used as data field in a process to represent transfer of value

The asset management system consists of following functionalities:



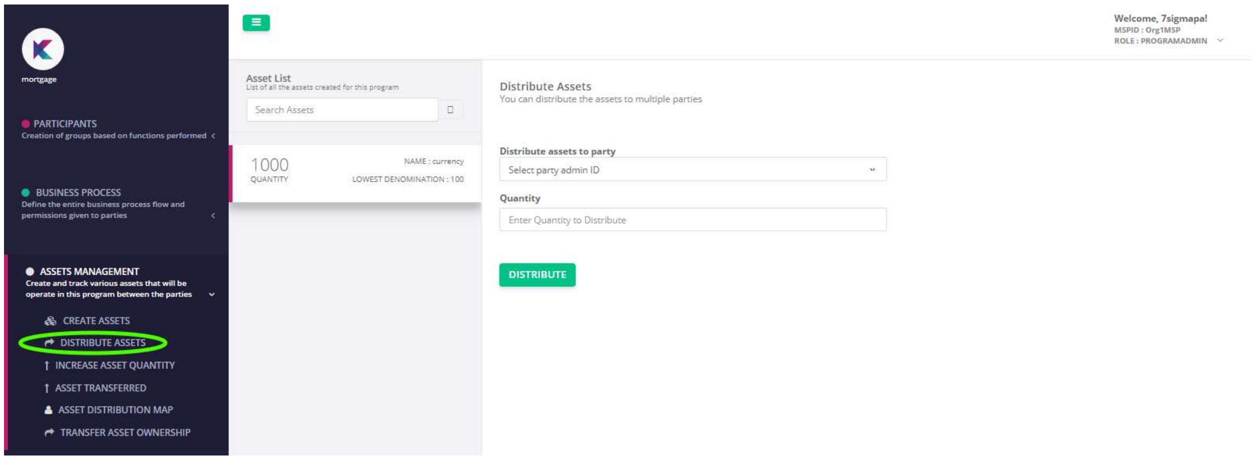
Create Assets
Set up different assets that will operate and assign the maximum units in circulation within this program.

Asset Name: Enter Asset Name
Eg: VC, INR
Asset Type: Select Asset Type
Quantity: Enter Assets Quantity
Eg: 100
Lowest Denomination: Enter Lowest Denomination

Creation of assets

- Expand the "**ASSET MANAGEMENT**" group and click on the "**CREATE ASSET**" tab
- On the Create Asset screen enter an "**Asset Name**". Entry of asset name should be unique. The asset name should be created in such a way that to use it as a reference becomes simple
- Select the asset type
- Select the quantity of Asset that you are required to be created
- Enter the lowest denomination divisor to which a particular asset can be handled. For example if lowest denomination of an asset is specified as 100, then each asset created has 100 parts to each and can be transmitted across in the lowest value of 0.01
- You can also switch the creation of additional asset tab to "on" in case you want to introduce more asset of similar type during the course of the program

Distribution of Asset

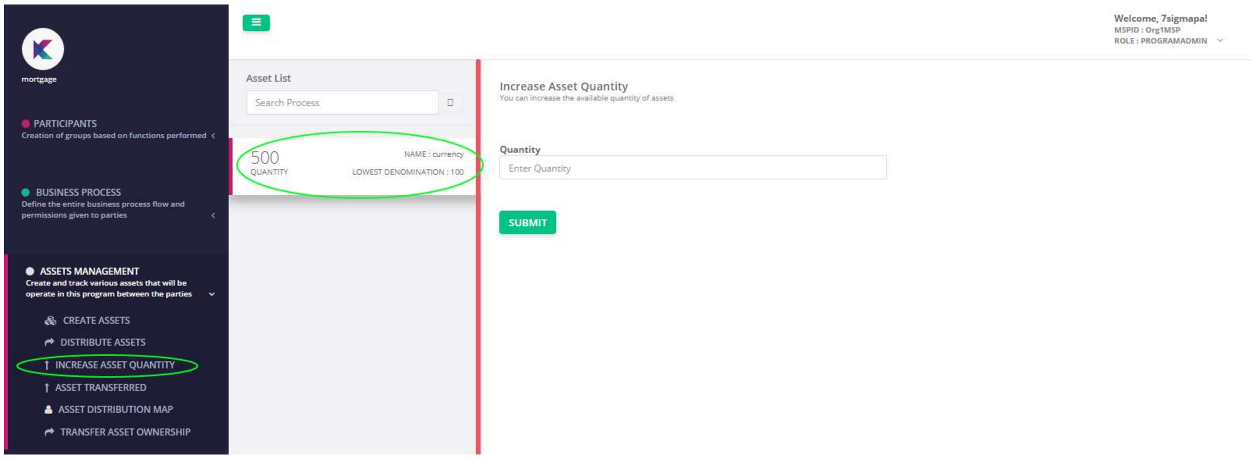

mortgage
PARTICIPANTS
Creation of groups based on functions performed
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
CREATE ASSETS
DISTRIBUTE ASSETS
INCREASE ASSET QUANTITY
ASSET TRANSFERRED
ASSET DISTRIBUTION MAP
TRANSFER ASSET OWNERSHIP
Welcome, 7sigmapa!
MSPID : Org1MSP
ROLE : PROGRAMADMIN
Asset List
List of all the assets created for this program
Search Assets
1000
QUANTITY
NAME : currency
LOWEST DENOMINATION : 100
Distribute Assets
You can distribute the assets to multiple parties
Distribute assets to party
Select party admin ID
Quantity
Enter Quantity to Distribute
DISTRIBUTE

After the assets have been created the next step is to distribute it amongst the party admin.

To distribute the assets click on "**DISTRIBUTE ASSETS**" under Asset Management group and select the Asset that you want to distribute. The inputs that are required on the asset distribution screen are:

- Distribute Asset to party: Select the intended party admin that needs to receive the assets as per the program requirements
- Quantity: Enter the quantity that needs to be transferred to the selected part administrator. The entered quantity needs to be lesser than the available asset balance at that instance

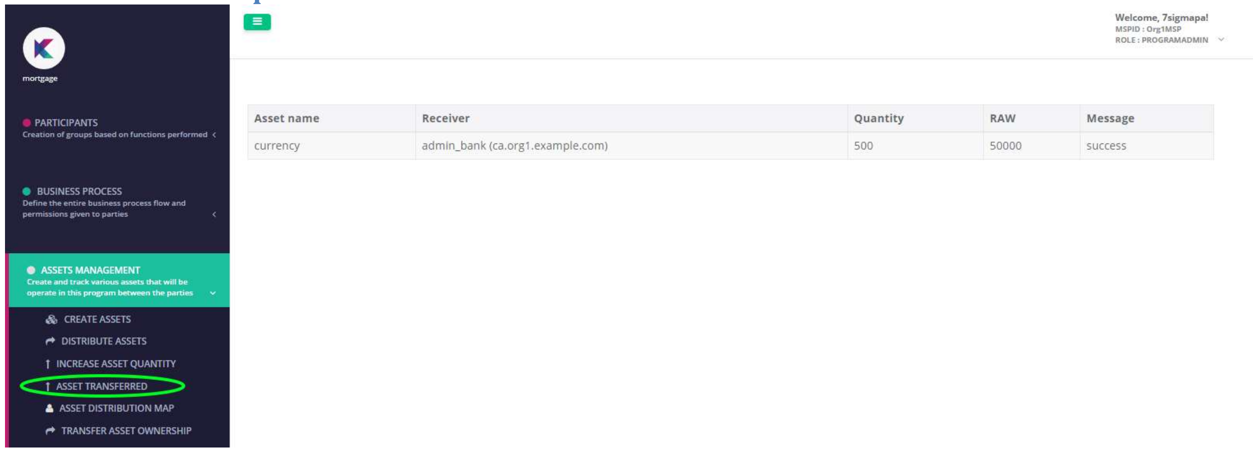
Increasing Asset Quantity:



During the course of execution of your use case, if addition of new assets to a created asset is required. You can use the function "**INCREASE ASSET QUANTITY**" that is available in the asset management group in the functional menu bar. The steps to follow post the selection of "**INCREASE ASSET QUANTITY**" tab:

- Select required asset that needs introduction of additional quantity
- Enter the quantity that you need to add and click on "SUBMIT" button

Asset Transfers Report:



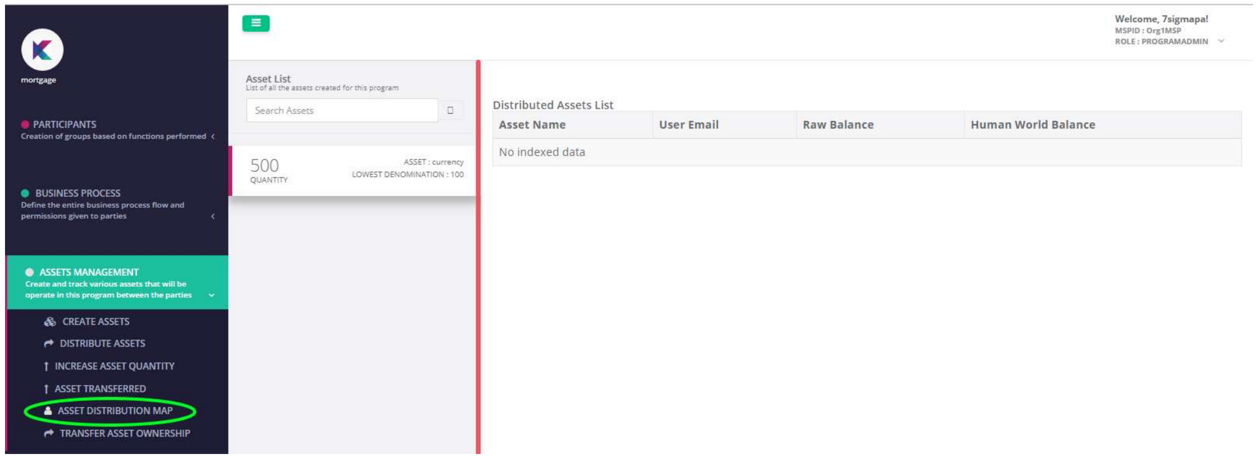
Asset name	Receiver	Quantity	RAW	Message
currency	admin_bank (ca.org1.example.com)	500	50000	success

This is report provided by KrypCore application that gives you a listing of asset transfers between various users that have taken place till date within the defined ecosystem of the use case or program.

The report provides the following information:

Asset Name: The name of the asset as defined during the creation of the asset
Receiver: The receiver of the asset
Quantity: The actual quantity of asset that was transferred
RAW: The inherent mathematical quantity of asset that was transferred, for example the asset that was created had the lowest denomination of 100, thus for an actual quantity of 1000 assets a RAW quantity of (actual quantity X lowest denomination i.e. 1000 X 100 = 100000) asset gets created, thus when you move 500 assets to an user, internally 500*100 assets gets transferred to the receiver
Message: States the status of the asset transfer transaction

Asset Distribution Mapping:

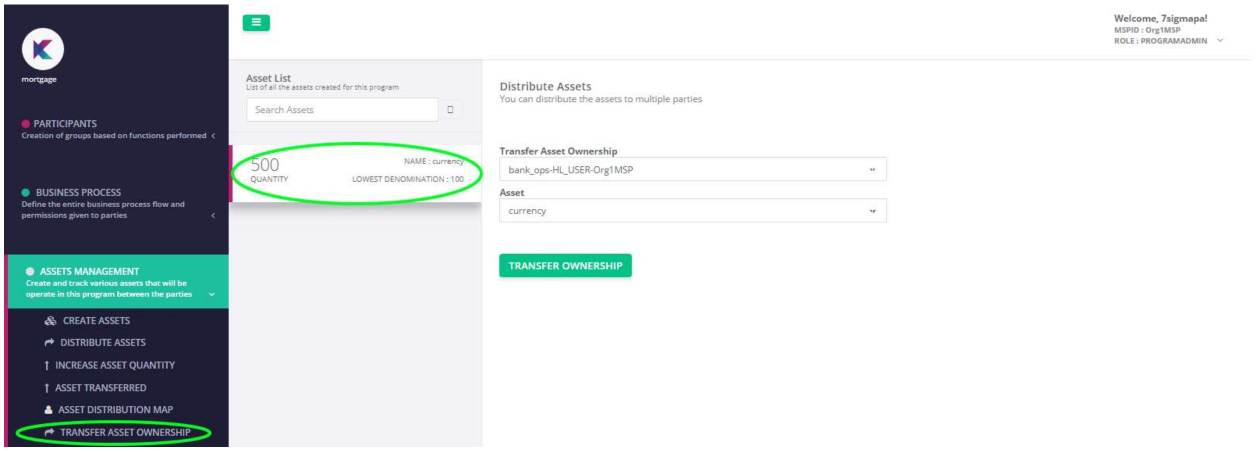

mortgage
PARTICIPANTS
Creation of groups based on functions performed
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
CREATE ASSETS
DISTRIBUTE ASSETS
INCREASE ASSET QUANTITY
ASSET TRANSFERRED
ASSET DISTRIBUTION MAP
TRANSFER ASSET OWNERSHIP
Welcome, 7sigmapa!
MSPID : Org1MSP
ROLE : PROGRAMADMIN
Asset List
List of all the assets created for this program
Search Assets

500
QUANTITY
ASSET : currency
LOWEST DENOMINATION : 100

Distributed Assets List

Asset Name	User Email	Raw Balance	Human World Balance
No indexed data			

Transferring Asset Ownership:


Welcome, 7sigmapa!
MSPID : Org1MSP
ROLE : PROGRAMADMIN

Asset List
List of all the assets created for this program

500 QUANTITY
NAME : currency
LOWEST DENOMINATION : 100

Distribute Assets
You can distribute the assets to multiple parties

Transfer Asset Ownership: bank_ops-HL_USER-Org1MSP
Asset: currency

mortgage
PARTICIPANTS
Creation of groups based on functions performed
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
CREATE ASSETS
DISTRIBUTE ASSETS
INCREASE ASSET QUANTITY
ASSET TRANSFERRED
ASSET DISTRIBUTION MAP
TRANSFER ASSET OWNERSHIP
Search Assets
TRANSFER OWNERSHIP

During the course of running the program it may be required that the ownership of asset gets changed from the current recipient to a different user at a program definition level. This can be achieved by using "**TRANSFER ASSET OWNERSHIP**" function. To change the ownership of the asset the following steps needs to be followed:

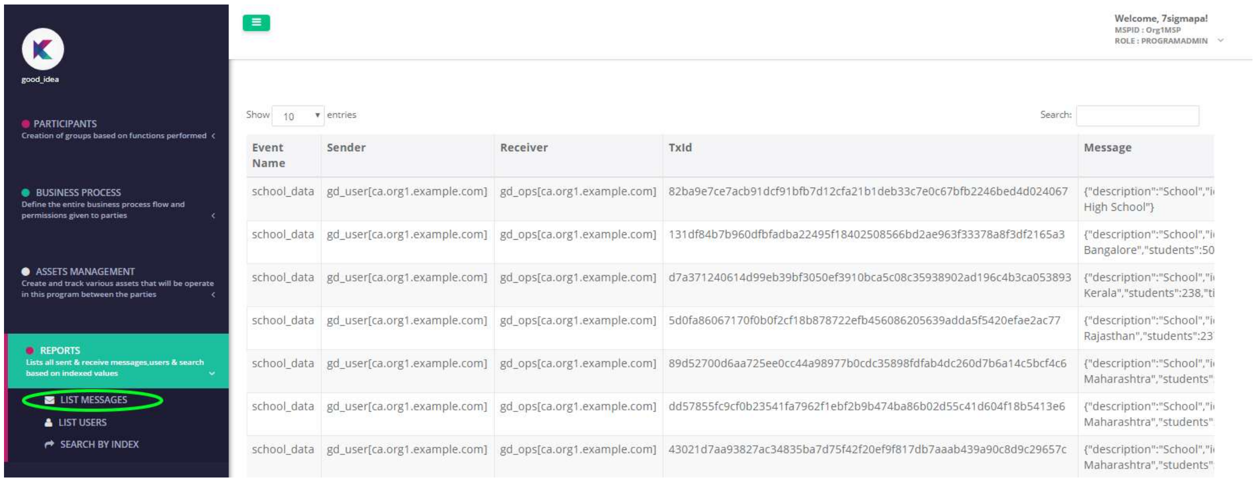
- Click on "**ASSETS MANAGEMENT**" to expand it
- Click on "**TRANSFER ASSETOOWNERSHIP**" tab
- Click on the asset where an asset ownership changes is required, this opens a form:
  - ➢ Select the user you want to make the new owner of the asset in "**Transfer Asset Ownership** "field
  - ➢ Select the asset from the displayed asset list

REPORTS:

KrypCore provides 3 reports at a program administrator level. These reports are generated with real time data and gives information about various facets of the program. The lists of reports made available by KrypCore application are:

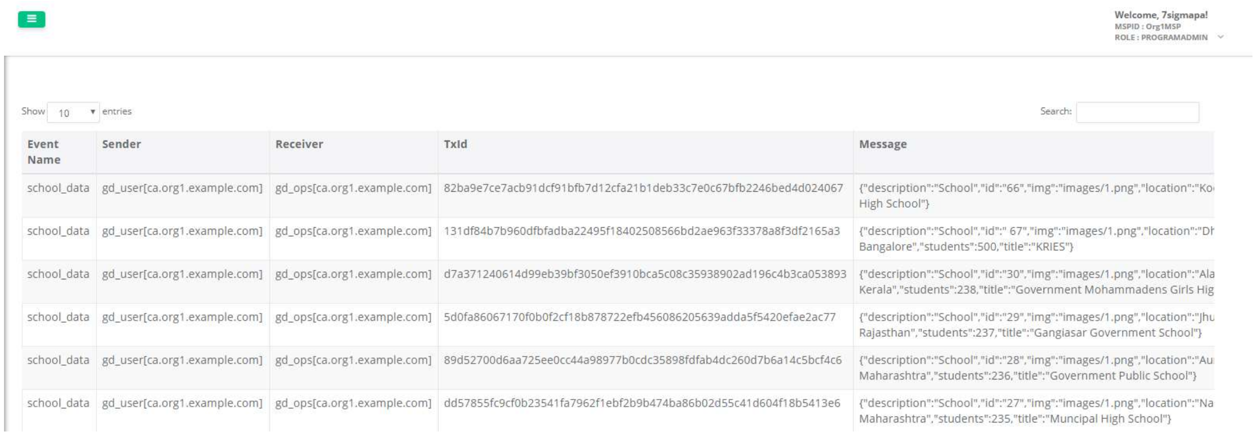
LIST MESSAGES:

This report provides a list of all the invoke transaction that has taken place till date in a program. To access "**LIST MESSAGES**" report, click on "**REPORTS**" on the functional menu bar to expand it showcasing the list of available reports and select on "**LIST MESSAGES**" tab:


good_idea
PARTICIPANTS
Creation of groups based on functions performed
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORTS
Lists all sent & receive messages,users & search based on indexed values
LIST MESSAGES
LIST USERS
SEARCH BY INDEX
Welcome, 7sigmapa!
MSPID : Org1MSP
ROLE : PROGRAMADMIN
Show 10 entries
Search:

Event Name	Sender	Receiver	TxId	Message
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	82ba9e7ce7acb91dcf91bfb7d12cfa21b1deb33c7e0c67bfb2246bed4d024067	{"description":"School","i High School"}
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	131df84b7b960dfbfadba22495f18402508566bd2ae963f33378a8f3df2165a3	{"description":"School","i Bangalore","students":50
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	d7a371240614d99eb39bf3050ef3910bca5c08c35938902ad196c4b3ca053893	{"description":"School","i Kerala","students":238,"ti
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	5d0fa86067170f0b0f2cf18b878722efb456086205639adda5f5420efae2ac77	{"description":"School","i Rajasthan","students":23
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	89d52700d6aa725ee0cc44a98977b0cdc35898fdfab4dc260d7b6a14c5bcf4c6	{"description":"School","i Maharashtra","students"
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	dd57855fc9cf0b23541fa7962f1ebf2b9b474ba86b02d55c41d604f18b5413e6	{"description":"School","i Maharashtra","students"
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	43021d7aa93827ac34835ba7d75f42f20ef9f817db7aaab439a90c8d9c29657c	{"description":"School","i Maharashtra","students"

On clicking on "List Messages" a page opens providing the details of all the transactions that has been recorded in the blockchain till date.



Event Name	Sender	Receiver	TxId	Message
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	82ba9e7ce7acb91dcf91bfb7d12cfa21b1deb33c7e0c67bfb2246bed4d024067	{"description":"School","id":"66","img":"images/1.png","location":"Ko High School"}
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	131df84b7b960dfbfadba22495f18402508566bd2ae963f33378a8f3df2165a3	{"description":"School","id":" 67","img":"images/1.png","location":"Dh Bangalore","students":500,"title":"KRIES"}
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	d7a371240614d99eb39bf3050ef3910bca5c08c35938902ad196c4b3ca053893	{"description":"School","id":"30","img":"images/1.png","location":"Ala Kerala","students":238,"title":"Government Mohammadens Girls Hig
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	5d0fa86067170f0b0f2cf18b878722efb456086205639adda5f5420efae2ac77	{"description":"School","id":"29","img":"images/1.png","location":"Jhu Rajasthan","students":237,"title":"Gangiasar Government School"}
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	89d52700d6aa725ee0cc44a98977b0cdc35898fdfab4dc260d7b6a14c5bcf4c6	{"description":"School","id":"28","img":"images/1.png","location":"Au Maharashtra","students":236,"title":"Government Public School"}
school_data	gd_user[ca.org1.example.com]	gd_ops[ca.org1.example.com]	dd57855fc9cf0b23541fa7962f1ebf2b9b474ba86b02d55c41d604f18b5413e6	{"description":"School","id":"27","img":"images/1.png","location":"Na Maharashtra","students":235,"title":"Muncipal High School"}

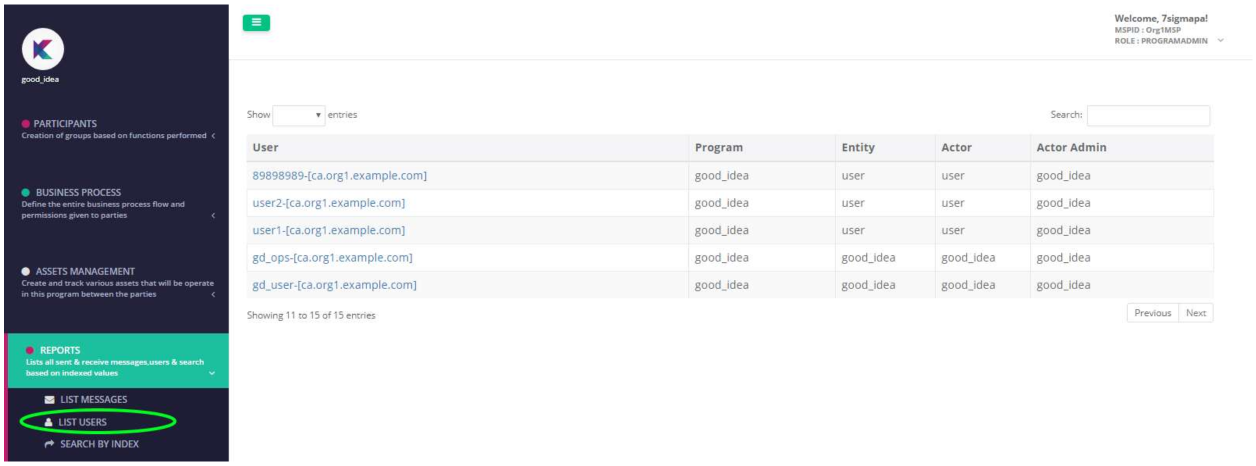
Details of the report parameters are discussed below:

- **Event Name**: Shows the event/process ID corresponding to the information displayed in that row
- **Sender**: Relates to the participating user who has initiated the event/process

- **Receiver**: Provides information about the recipient user of the referred event/process
- **TxId**: Gives the transaction Id that was generated at a blockchain protocol level after the event/process got successfully executed
- **Message**: Furnish details on the data /payload or the parameter value that got recorded into the blockchain network after a successful execution of the process step

LIST USERS:

This report provides details on the user hierarchy that is currently mapped against the respective program. To get the report click on the "**LIST USERS**" tab within the "**REPORTS**" group:


good_idea
PARTICIPANTS
Creation of groups based on functions performed
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORTS
Lists all sent & receive messages,users & search based on indexed values
LIST MESSAGES
LIST USERS
SEARCH BY INDEX
Welcome, 7sigmapa!
MSPID : Org1MSP
ROLE : PROGRAMADMIN
Show entries
Search:
Previous
Next

User	Program	Entity	Actor	Actor Admin
89898989-[ca.org1.example.com]	good_idea	user	user	good_idea
user2-[ca.org1.example.com]	good_idea	user	user	good_idea
user1-[ca.org1.example.com]	good_idea	user	user	good_idea
gd_ops-[ca.org1.example.com]	good_idea	good_idea	good_idea	good_idea
gd_user-[ca.org1.example.com]	good_idea	good_idea	good_idea	good_idea

Showing 11 to 15 of 15 entries

The report contains the following details:
User: The end user who has the execution rights of selected events that can either invoke or query data to/from the blockchain network or the synced data lake
Program: Provides the name of the use case with which the user is operating
Entity: Provides the participant ID that the respective user belongs to
Actor: Provides the party ID that has allowed the users to be mapped and has provided appropriate permissions on event execution to the referred user
Actor Admin: The administrator that represents the specific Actor or party in the eco-system

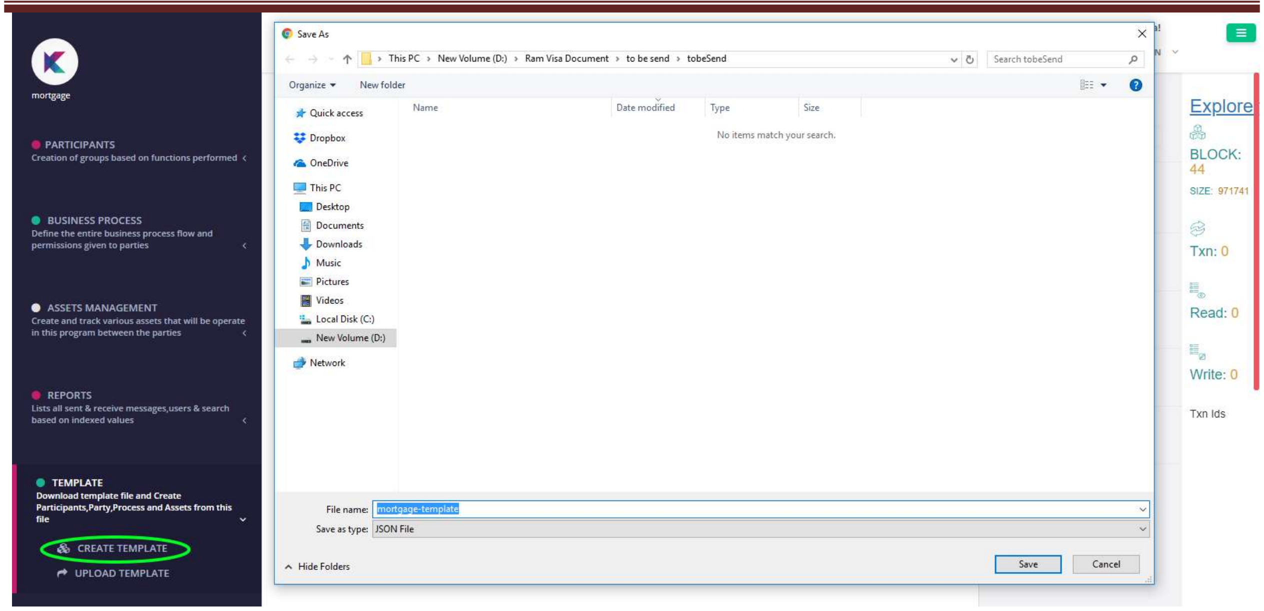
TEMPLATE:

Another functionality provided by KrypCore application allowing the program administrator to download template file of the mapped program allowing for transferring the file to a different blockchain network/setup

There are two functionalities provided under the "**TEMPLATE"** group in the functional menu bar:

CREATE TEMPLATE:

Allow you to create template file of the program, the JSON file thus created contains the latest state or the program definition including the mapped users, created participants and parties as well as the process /events defined along with the business logic. Also note that the template files enables you to create the overall structure and participants of the respective program and not the transaction data captured during event executions with the purview of the respective program


mortgage
PARTICIPANTS
Creation of groups based on functions performed
BUSINESS PROCESS
Define the entire business process flow and permissions given to parties
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORTS
Lists all sent & receive messages,users & search based on indexed values
TEMPLATE
Download template file and Create Participants,Party,Process and Assets from this file
CREATE TEMPLATE
UPLOAD TEMPLATE
Save As
This PC › New Volume (D:) › Ram Visa Document › to be send › tobeSend
Search tobeSend
Organize
New folder
Quick access
Dropbox
OneDrive
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
New Volume (D:)
Network
Name
Date modified
Type
Size
No items match your search.
File name: mortgage-template
Save as type: JSON File
Hide Folders
Save
Cancel
Explore
BLOCK: 44
SIZE: 971741
Txn: 0
Read: 0
Write: 0
Txn Ids

You can save the template file in the requisite storage are by providing the location path, followed by saving the file

UPLOAD TEMPLATE: Copy the content of template downloaded from other instances and past it in the text box provide. This action will create the business application in the current program / instance.

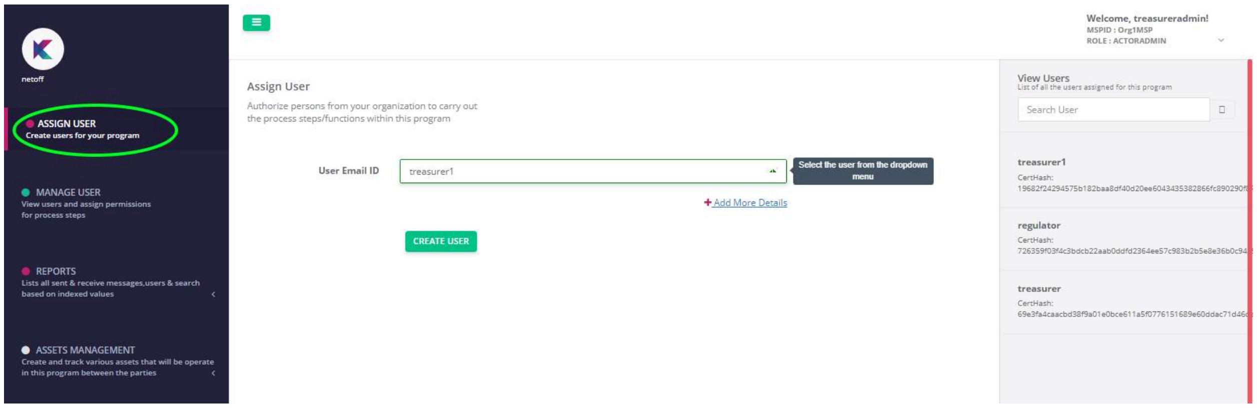
INDEXED: Not Applicable in this version

MAPPING END USERS AND GIVING EVENT/PROCESS PERMISSION

After a program is defined/ mapped by the program admin, the next step is for the respective party administrators to map respective end users into the program purview and assign requisite permissions of the event/process. These events/ process are in lieu of program manager giving send and receive permission to the respective participant group they belong to

To start the process of mapping the end users, the respective party admin logs in using its login credentials. On successful validation the party administrator is allowed to carry out the following processes:

ASSIGN USER:


netoff
ASSIGN USER
Create users for your program
MANAGE USER
View users and assign permissions for process steps
REPORTS
Lists all sent & receive messages,users & search based on indexed values
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
Welcome, treasureradmin!
MSPID : Org1MSP
ROLE : ACTORADMIN
Assign User
Authorize persons from your organization to carry out the process steps/functions within this program
User Email ID
treasurer1
Select the user from the dropdown menu
+ Add More Details
CREATE USER
View Users
List of all the users assigned for this program
Search User

treasurer1
CertHash:
19682f24294575b182baa8df40d20ee6043435382866fc890290f

regulator
CertHash:
726359f03f4c3bdcb22aab0ddfd2364ee57c983b2b5e8e36b0c94

treasurer
CertHash:
69e3fa4caacbd38f9a01e0bce611a5f0776151689e60ddac71d46

The first step is to assign users from the list of user created during the user registration stage

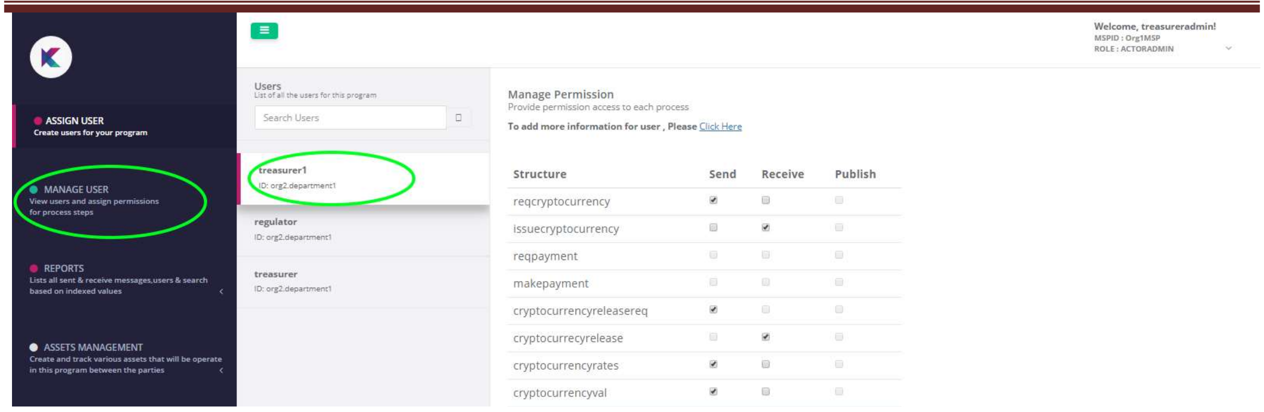
Note: One end user can only be assigned by one party admin to carry out the events it inherits

- To assign user the party admin has to click on "**ASSIGN USER**" tab
- In the Assign User section party admin chooses from the listed users and clicks on "**CREATE USER**" button

During user creation leg the selected user gets mapped to the respective Party Admin User

MANAGE USER

After mapping of the requisite users, the next step carried out by the party administrator is to assign send or receive permissions to each of the end users created. Only the accessible events/ processes that the party administrator has can be further provided to each of the users it has mapped


Welcome, treasureradmin!
MSPID : Org1MSP
ROLE : ACTORADMIN

ASSIGN USER
Create users for your program

MANAGE USER
View users and assign permissions for process steps

REPORTS
Lists all sent & receive messages,users & search based on indexed values

ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties

Users
List of all the users for this program

Search Users

treasurer1
ID: org2.department1

regulator
ID: org2.department1

treasurer
ID: org2.department1

Manage Permission
Provide permission access to each process

To add more information for user , Please Click Here

Structure	Send	Receive	Publish
reqcryptocurrency	✓	☐	☐
issuecryptocurrency	☐	✓	☐
reqpayment	☐	☐	☐
makepayment	☐	☐	☐
cryptocurrencyreleasereq	✓	☐	☐
cryptocurrecyrelease	☐	✓	☐
cryptocurrencyrates	✓	☐	☐
cryptocurrencyval	✓	☐	☐

To start providing the permissions of the event/process to the end user, the party admin needs to:

- Click on the "**MANAGE USER**" tab on the functional menu bar
- Select one the mapped user
- On selection of the user, a table with all the mapped events in the program is listed on the right hand side of the screen
- :where the send /receive tick boxes are highlighted
- After assigning the requisite permission to the user, party administrator then needs to click on "**SAVE PERMISSION**" button to enable recording of the assignment in the blockchain. On successful commit of the transaction, transaction Id from the blockchain protocol is displayed on the screen

REPORTS

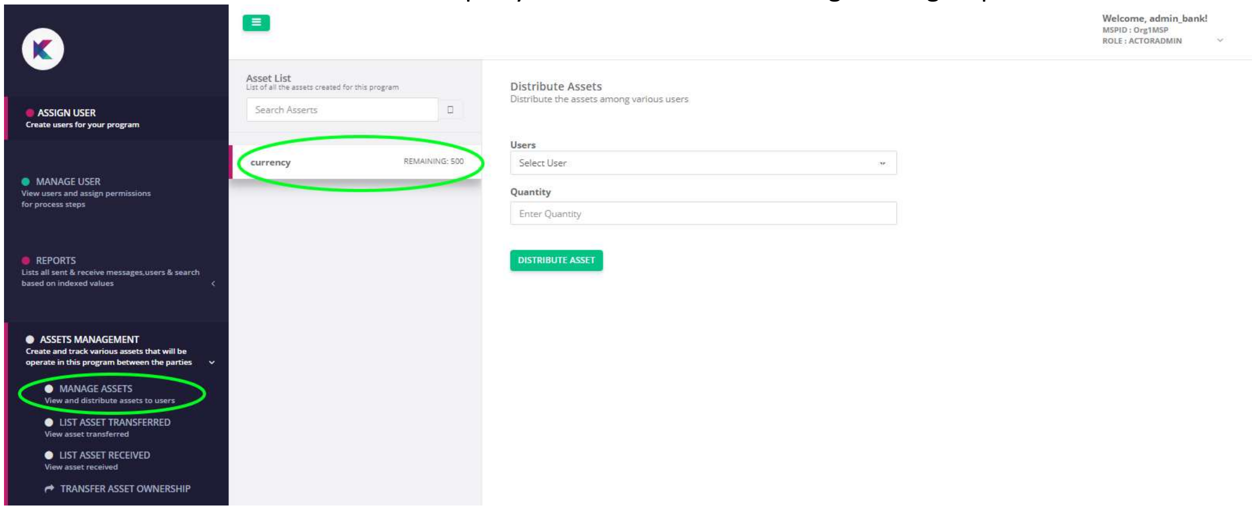
Party administrators also have some reports available to them. These reports are:

LIST MESSAGES: Provides the details of all the successful invoke transaction carried out by the users mapped to the respective party administrator

LIST USERS: Lists down **the** users mapped to the querying party administrator

ASSET MANAGEMENT

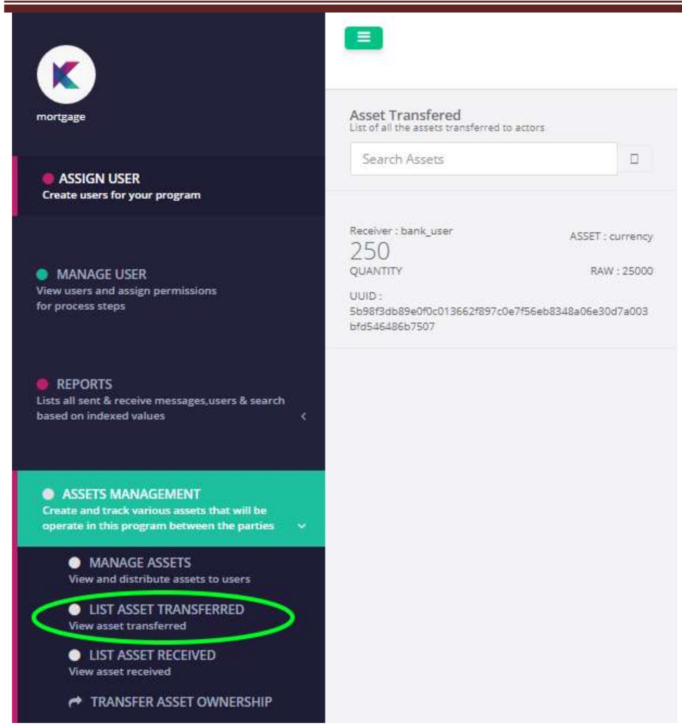
Other than the mapping of the users the next process that a party administrator needs to carry out is to manage assets that it may have received from the program administrator.
The various functionalities available to party admin under asset management group are:


Welcome, admin_bank!
MSPID : Org1MSP
ROLE : ACTORADMIN
ASSIGN USER
Create users for your program
MANAGE USER
View users and assign permissions for process steps
REPORTS
Lists all sent & receive messages,users & search based on indexed values
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
MANAGE ASSETS
View and distribute assets to users
LIST ASSET TRANSFERRED
View asset transferred
LIST ASSET RECEIVED
View asset received
TRANSFER ASSET OWNERSHIP
Asset List
List of all the assets created for this program
Search Asserts
currency
REMAINING: 500
Distribute Assets
Distribute the assets among various users
Users
Select User
Quantity
Enter Quantity
DISTRIBUTE ASSET

MANAGE ASSETS: This functionality aids the party admin to distribute the assets that it has received from the program manager to the end users that has been mapped by him.

- To distribute the assets party administrator needs to click on the “**MANAGE USER**” tab
- Select one of the users mapped to the respective party admin
- Enter the quantity of asset to be allocated

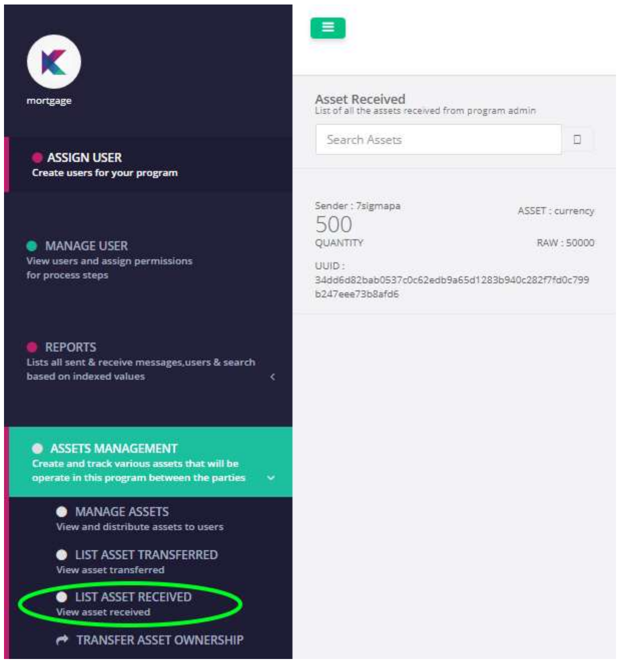
LIST ASSET TRANSFERRED: This functionality provide a report of all the asset transfer transaction done by the Party Administrator to its users. Each of these transactions is uniquely identifiable against the UUID (transaction identity provided by blockchain network on successful commit)


mortgage
ASSIGN USER
Create users for your program
MANAGE USER
View users and assign permissions for process steps
REPORTS
Lists all sent & receive messages,users & search based on indexed values
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
MANAGE ASSETS
View and distribute assets to users
LIST ASSET TRANSFERRED
View asset transferred
LIST ASSET RECEIVED
View asset received
TRANSFER ASSET OWNERSHIP
Search Assets

Asset Transfered
List of all the assets transferred to actors

Receiver : bank_user
ASSET : currency
250
QUANTITY
RAW : 25000
UUID :
5b98f3db89e0f0c013662f897c0e7f56eb8348a06e30d7a003
bfd546486b7507

LIST ASSET RECEIEVD: This functionality provides a report on the transactions that were executed by the program administrator to transfer asset to the Party Administrator. Unique UUID refers to individual asset transfer transactions carried out


mortgage
ASSIGN USER
Create users for your program
MANAGE USER
View users and assign permissions for process steps
REPORTS
Lists all sent & receive messages,users & search based on indexed values
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
MANAGE ASSETS
View and distribute assets to users
LIST ASSET TRANSFERRED
View asset transferred
LIST ASSET RECEIVED
View asset received
TRANSFER ASSET OWNERSHIP
Search Assets

Asset Received
List of all the assets received from program admin

Sender : 7sigmapa
ASSET : currency
500
QUANTITY
RAW : 50000
UUID :
34dd6d82bab0537c0c62edb9a65d1283b940c282f7fd0c799
b247eee73b8afd6

TRANSFER ASSET OWNERSHIP: Control of Re-Issuance can be transferred to other users who is part the current program.

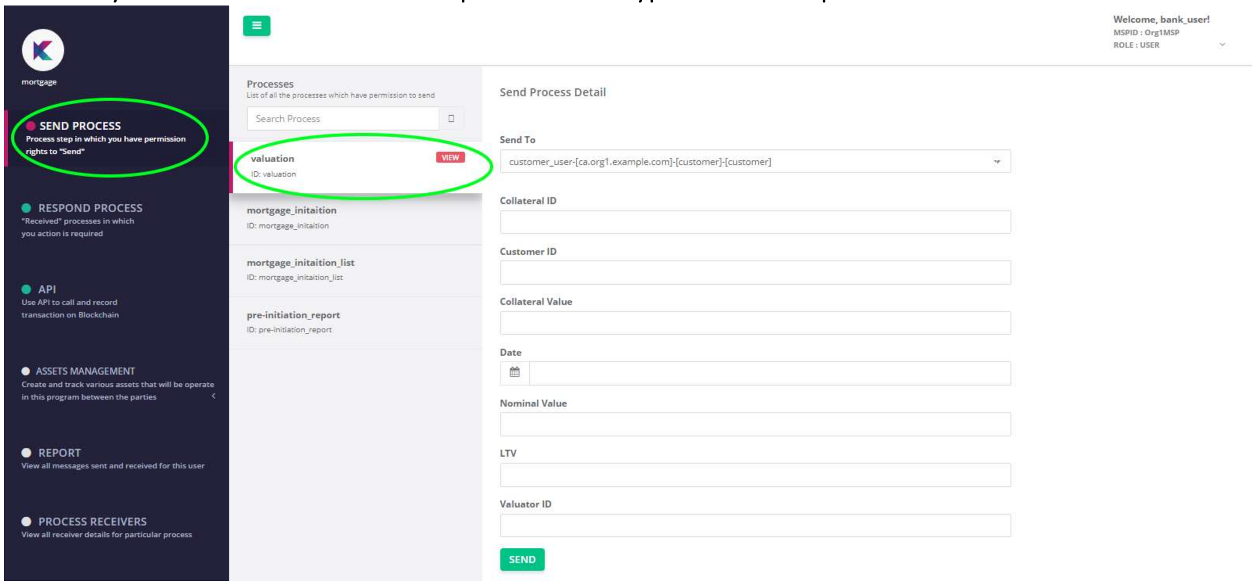
END USER EXECUTION OF THE EVENTS/PROCESSES

After all the party administrators in the use case ecosystem have completed mapping the relevant users to the requisite events/processes, one can login using the user credentials to execute the mapped events.

Krypcore application automatically creates forms based on the parameters specified in a event/process as well as generates RESTful API relevant to that particular event.

To access the end user portal one needs to login with the end users credentials linked to the program.

We briefly describe the functionalities provided to a typical end user portal:


mortgage
SEND PROCESS
Process step in which you have permission rights to "Send"
RESPOND PROCESS
"Received" processes in which you action is required
API
Use API to call and record transaction on Blockchain
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORT
View all messages sent and received for this user
PROCESS RECEIVERS
View all receiver details for particular process
Processes
List of all the processes which have permission to send
Search Process
VIEW
SEND

Welcome, bank_user!
MSPID : Org1MSP
ROLE : USER

Process	ID
valuation	ID: valuation
mortgage_initaition	ID: mortgage_initaition
mortgage_initaition_list	ID: mortgage_initaition_list
pre-initiation_report	ID: pre-initiation_report

Send Process Detail

Send To: customer_user-[ca.org1.example.com]-[customer]-[customer]

Collateral ID

Customer ID

Collateral Value

Date

Nominal Value

LTV

Valuator ID

SEND PROCESS

List down all the events that the logged in user has send permission for. Typically only the list of transactions where either the bind rule is specified as “**None**” /”**REPORT_WALLET**” or “**REPORT_LOCAL**”

The process steps to be followed by user to execute invoke or report event/process is appended below:

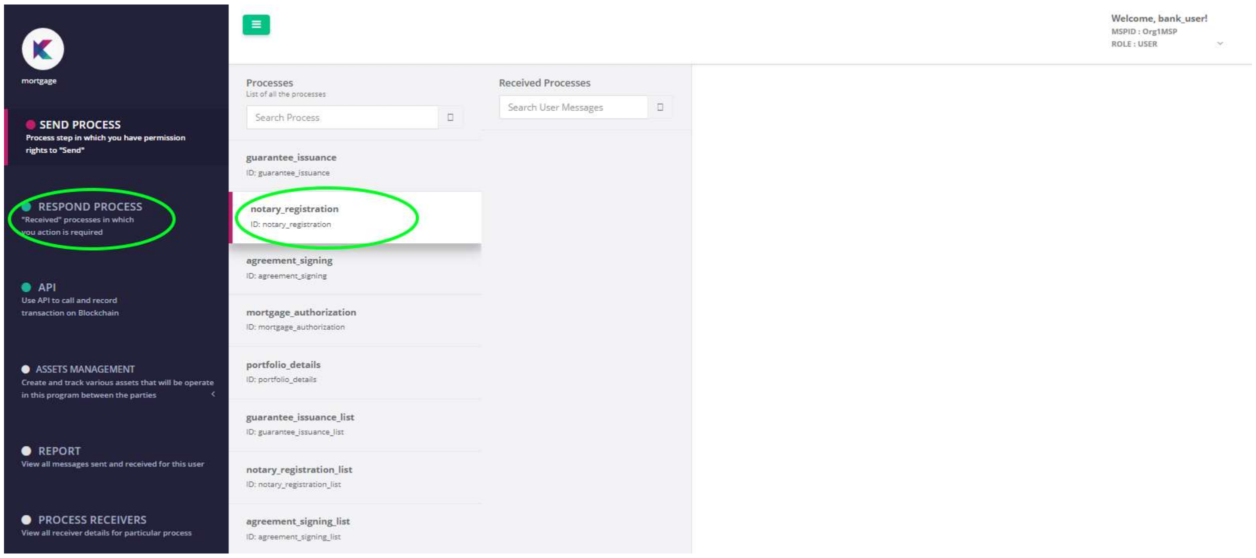
- Select the event that needs to be executed
- On selecting the event, a form with the parameters lists defined during the designing of the event gets automatically populated
- Select the requisite receiver from the list of receivers in the list box against the “**Send To**” label

Note: Only users which have receive permission against this event will be listed

- Ensure that the sender and receiver are not the same
- Fill in the data and click on “**SEND**” button

- On successful submission of invoke transactions you will get a SUCCESS message with the UUID from the blockchain network
- In case of reports, based on the way you have coded your report either you can see the report contents in the same screen listed in a table below the form or you need to go the respective browser console and retrieve the report from the console window which will be in form of a JSON

RESPOND PROCESS


Welcome, bank_user!
MSPID : Org1MSP
ROLE : USER
mortgage
SEND PROCESS
Process step in which you have permission rights to "Send"
RESPOND PROCESS
"Received" processes in which you action is required
API
Use API to call and record transaction on Blockchain
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORT
View all messages sent and received for this user
PROCESS RECEIVERS
View all receiver details for particular process
Processes
List of all the processes
Search Process
Received Processes
Search User Messages
guarantee_issuance
ID: guarantee_issuance
notary_registration
ID: notary_registration
agreement_signing
ID: agreement_signing
mortgage_authorization
ID: mortgage_authorization
portfolio_details
ID: portfolio_details
guarantee_issuance_list
ID: guarantee_issuance_list
notary_registration_list
ID: notary_registration_list
agreement_signing_list
ID: agreement_signing_list

In case while defining an event you have chosen any one the below listed bind rules:

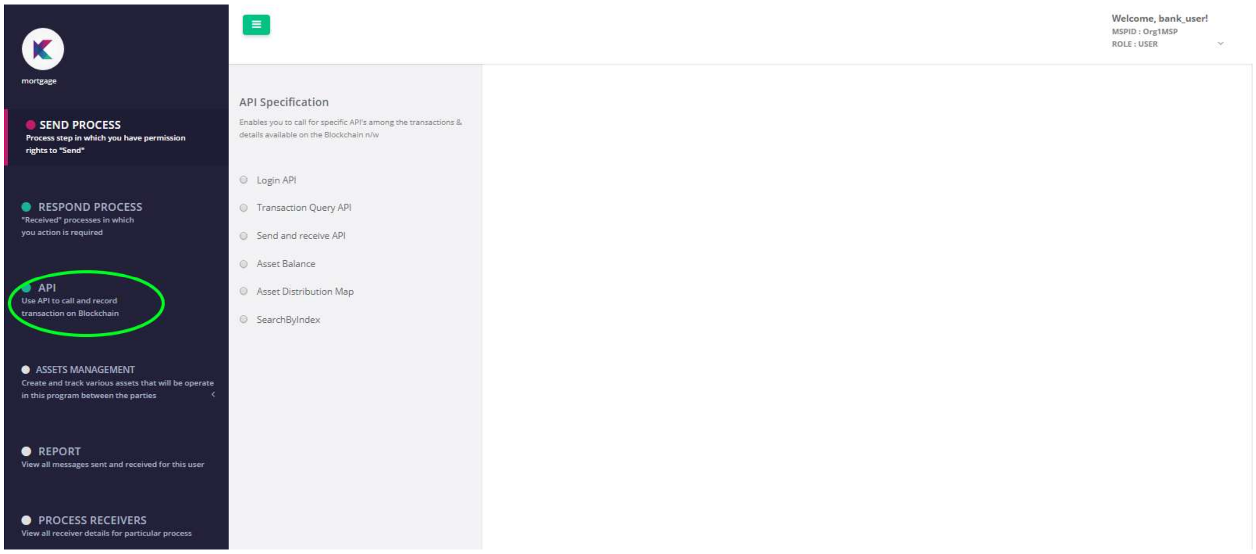
- Spend referred message: (Invoke)
- Spend referred message and reply to sender only:
- Refer a message: (Invoke)

You need to click on "**RESPOND PROCESS**" to execute such events

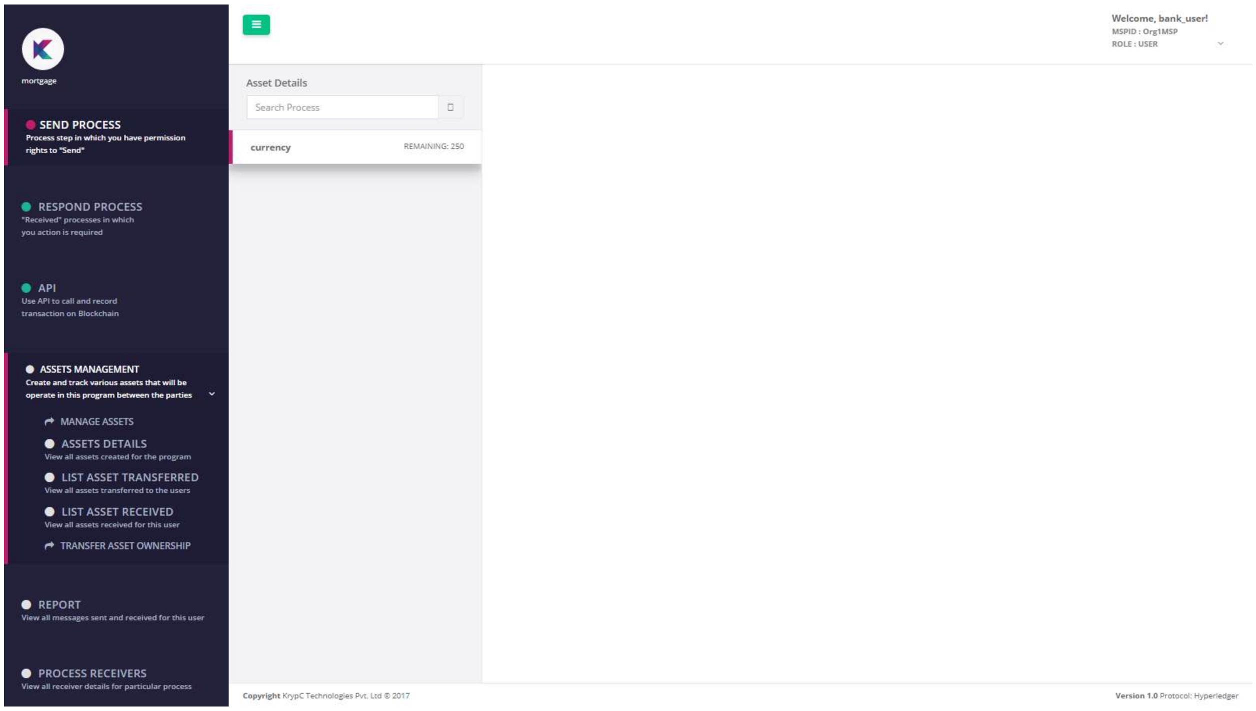
- Respond process will list down a list of predecessor events that needs to be executed before the event with the any of the above bind rules can be executed
- Click on one of the processes/events
- On selecting the event another form pops up showing the successor event with input fields against the parameters that is associated with the successor event
- Fill in the form with requisite details and click on "SEND" button
- On successful submission of invoke transactions you will get a SUCCESS message with the UUID from the blockchain network

API:

Users would be able to see the list of APIs along with payload specification. Using this information, existing application can interface with this blockchain application. There are different categories of APIs as shown in the below screen shot.


mortgage
SEND PROCESS
Process step in which you have permission rights to "Send"
RESPOND PROCESS
"Received" processes in which you action is required
API
Use API to call and record transaction on Blockchain
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORT
View all messages sent and received for this user
PROCESS RECEIVERS
View all receiver details for particular process
Welcome, bank_user!
MSPID : Org1MSP
ROLE : USER
API Specification
Enables you to call for specific API's among the transactions & details available on the Blockchain n/w
Login API
Transaction Query API
Send and receive API
Asset Balance
Asset Distribution Map
SearchByIndex

ASSET MANAGEMENT:


mortgage
SEND PROCESS
Process step in which you have permission rights to "Send"
RESPOND PROCESS
"Received" processes in which you action is required
API
Use API to call and record transaction on Blockchain
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
MANAGE ASSETS
ASSETS DETAILS
View all assets created for the program
LIST ASSET TRANSFERRED
View all assets transferred to the users
LIST ASSET RECEIVED
View all assets received for this user
TRANSFER ASSET OWNERSHIP
REPORT
View all messages sent and received for this user
PROCESS RECEIVERS
View all receiver details for particular process
Welcome, bank_user!
MSPID : Org1MSP
ROLE : USER
Asset Details
Search Process
currency
REMAINING: 250
Copyright KrypC Technologies Pvt. Ltd © 2017
Version 1.0 Protocol: Hyperledger

ASSET DETAILS: List down the assets the current user is subscribed to

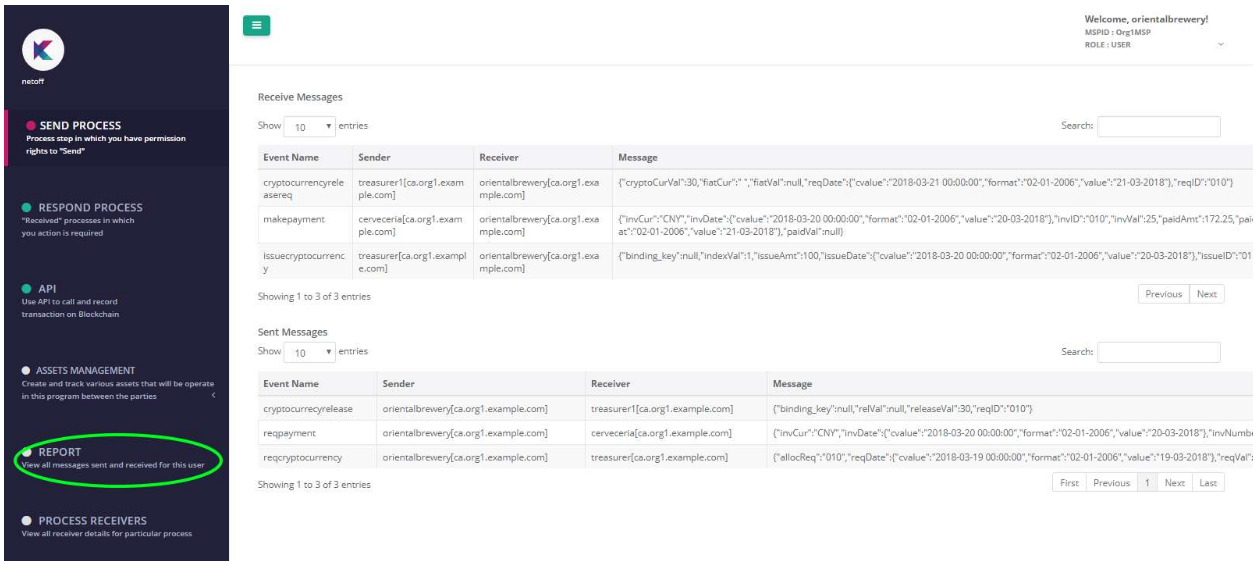
LIST ASSET TRANSFERRED: This functionality provide a report of all the asset transfer transaction done by the user to other users. Each of these transactions is uniquely identifiable against the UUID (transaction identity provided by blockchain network on successful commit)

LIST ASSET RECEIEVD: This functionality provides a report on the transactions that were executed by the party administrator to transfer asset to the logged in user. Unique UUID refers to individual asset transfer transactions carried out

TRANSFER ASSET OWNERSHIP: This functionality provides a report on the transactions (i.e. Transfer of control with respect to re-issuance of asset) that were executed by the user.

REPORT

The report functionality at an end user level provides report on the send transaction carried out by the user as well as receive transactions received by the user



netoff
SEND PROCESS
Process step in which you have permission rights to "Send"
RESPOND PROCESS
"Received" processes in which you action is required
API
Use API to call and record transaction on Blockchain
ASSETS MANAGEMENT
Create and track various assets that will be operate in this program between the parties
REPORT
View all messages sent and received for this user
PROCESS RECEIVERS
View all receiver details for particular process
Welcome, orientalbrewery!
MSPID : Org1MSP
ROLE : USER
Show 10 entries
Search:
Previous
Next
First
Last

Receive Messages

Event Name	Sender	Receiver	Message
cryptocurrencyreleasereq	treasurer1[ca.org1.example.com]	orientalbrewery[ca.org1.example.com]	{"cryptoCurVal":30,"fiatCur":" ","fiatVal":null,"reqDate":{"cvalue":"2018-03-21 00:00:00","format":"02-01-2006","value":"21-03-2018"},"reqID":"010"}
makepayment	cerveceria[ca.org1.example.com]	orientalbrewery[ca.org1.example.com]	{"invCur":"CNY","invDate":{"cvalue":"2018-03-20 00:00:00","format":"02-01-2006","value":"20-03-2018"},"invID":"010","invVal":25,"paidAmt":172.25,"pai at":"02-01-2006","value":"21-03-2018"},"paidVal":null}
issuecryptocurrency	treasurer[ca.org1.example.com]	orientalbrewery[ca.org1.example.com]	{"binding_key":null,"indexVal":1,"issueAmt":100,"issueDate":{"cvalue":"2018-03-20 00:00:00","format":"02-01-2006","value":"20-03-2018"},"issueID":"01

Showing 1 to 3 of 3 entries

Sent Messages

Event Name	Sender	Receiver	Message
cryptocurrecyrelease	orientalbrewery[ca.org1.example.com]	treasurer1[ca.org1.example.com]	{"binding_key":null,"relVal":null,"releaseVal":30,"reqID":"010"}
reqpayment	orientalbrewery[ca.org1.example.com]	cerveceria[ca.org1.example.com]	{"invCur":"CNY","invDate":{"cvalue":"2018-03-20 00:00:00","format":"02-01-2006","value":"20-03-2018"},"invNumb
reqcryptocurrency	orientalbrewery[ca.org1.example.com]	treasurer[ca.org1.example.com]	{"allocReq":"010","reqDate":{"cvalue":"2018-03-19 00:00:00","format":"02-01-2006","value":"19-03-2018"},"reqVal":

Showing 1 to 3 of 3 entries

The report contents are briefly discussed below:

Event name: Refers to the event/process ID that has been triggered in by the logged in user in "**Sent Messages**" section or has been received by the user "**Receive Message**" section

Sender: Refers to the user who has executed the respective event or process

Receiver: refers to the end user who has received the transaction executed by a Sender

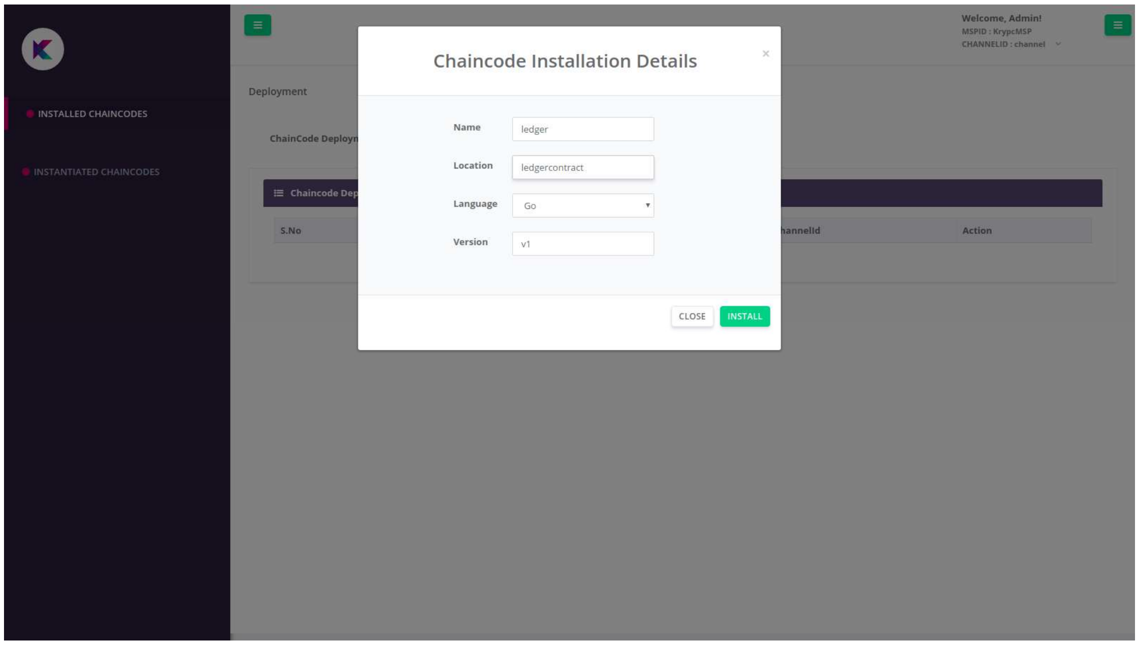
Messages: refers to the parameter value/payload data that got exchanged between the **Sender** and the **Receiver**

PROCESS RECIEVERS:

CHAIN CODE INSTALLATION

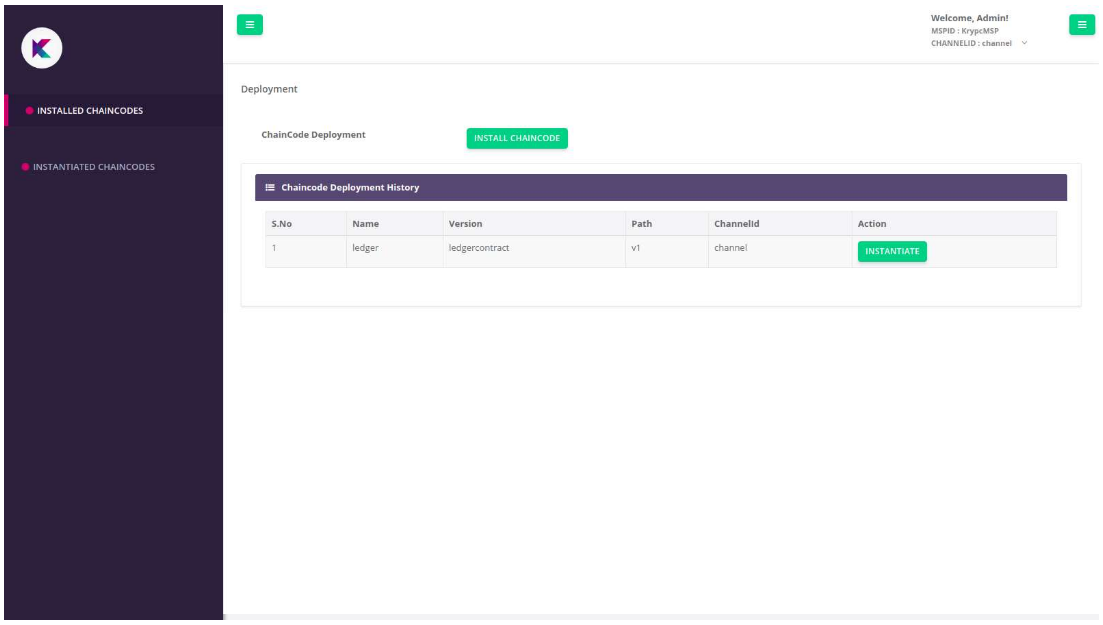
The chain code installation is done in the below window where we can specify four parameters.

- Name of the chaincode
- Path of the chaincode to be installed
- Language of the chaincode (flexibility to opt between GoLang, Java & JS)
- Specify the version of the chain code



Name	ledger
Location	ledgercontract
Language	Go
Version	v1

If the chaincode is being installed for the first time, it has to be instantiated. But if it is a modification to an existing chaincode then it will be an upgrade.

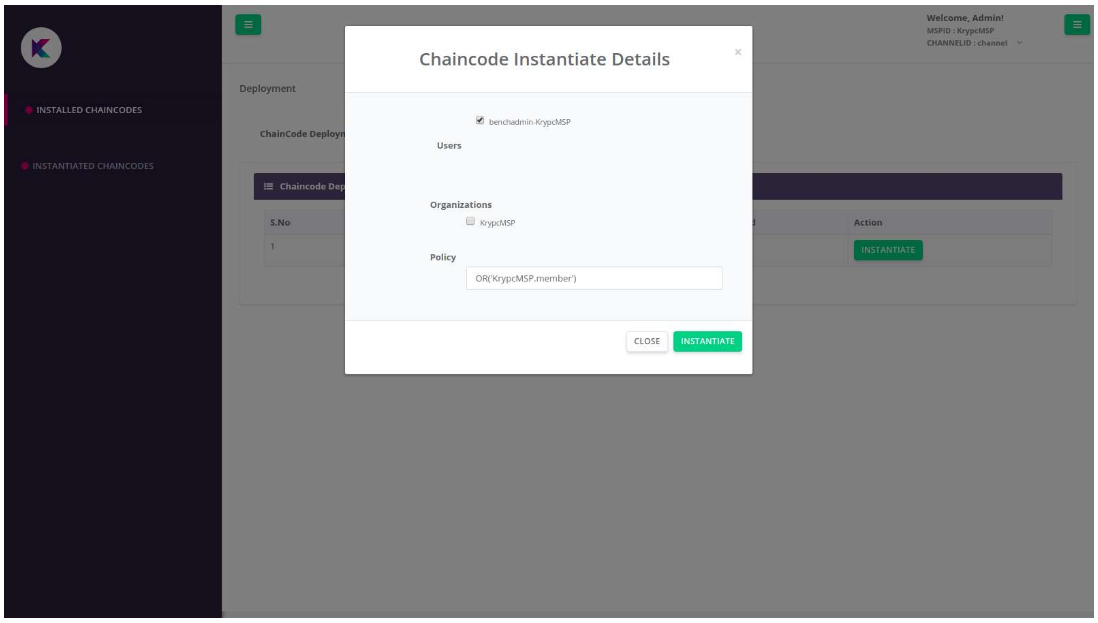


S.No	Name	Version	Path	ChannelId	Action
1	ledger	ledgercontract	v1	channel	INSTANTIATE

CHAIN CODE INSTANTIATION

The installed chain code can be instantiated in the below window where

- Bench admin selection can be done out of the existing users
- Organizations who are required to be a part of the endorsement policy can be selected (or)
- Endorsement policy of a channel can be defined



Chaincode Instantiate Details

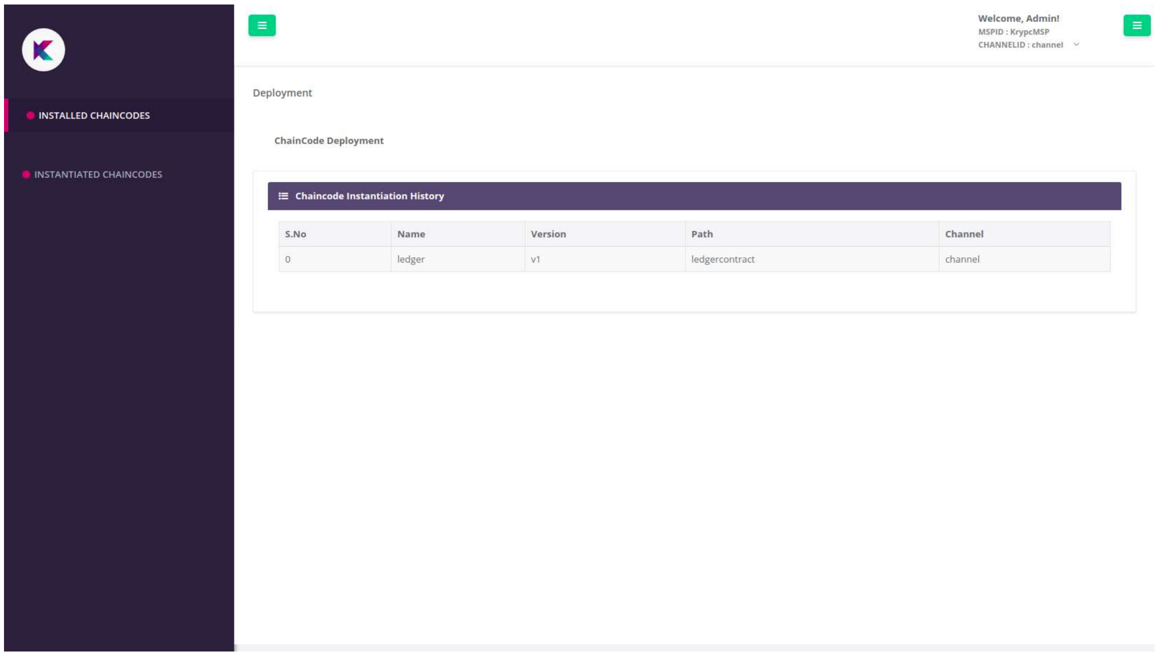
Users
☑ benchadmin-KrypcMSP

Organizations
☐ KrypcMSP

Policy
OR('KrypcMSP.member')

CLOSE INSTANTIATE

The list of instantiated chaincode can be viewed using the second tab.



Chaincode Instantiation History

S.No	Name	Version	Path	Channel
0	ledger	v1	ledgercontract	channel