TrustFactor Backoffice User Manual

Contents

Release Notes

This document is the user manual for the TrustFactor backoffice.

There are two main ways of integrating the TrustFactor Authentication Solution in your application – either through the TrustFactor App which is shared among customers and managed by SecuritySide or through Customized Applications – and choosing one or the other changes the level of access you have to the backoffice.

Customers integrating with the TrustFactor App will only have application-level permissions on the backoffice, whereas if you have a Customized Application you will have full owner permissions to the backoffice.

Throughout this document, we will mark the sections available only for Customized Application customers with the "CustomApps only" tag for clarity and simplicity.

Changelog - v4.8.0

- Add contract revalidation event
- Add contract list & details

Changelog - v4.7.0

- Add Authentication provider to application settings
- Change all dates strings to show according the ISO 8601

Changelog - v4.6.0

- Add T&C listing & details
- Add device location to devices list

Changelog - v4.5.0

- Event feed datails, show timestamp and CID values
- Virtual agents - Show a list of failed shared contracts

Changelog - v4.4.0

- Handle application callback errors (virtual agents)
- Change Error logs default 'Created At' filter value, to be 30 mins

- Show Error logs details in new tab
- Remove contract_key filter from transactions list
- Add button on successful creation of API Token to add event in calendar (to renew, containing all information about token)
- Implement lazy loading in get permissions request
- Handle error codes when api return error 500 (or other mapped status code)
- Add support to filter by CID on Error logs list

Changelog - v4.3.0

- Add Virtual Agents
- Implement scrollbar on sidebar
- Fix bug on change application via app selector

Changelog - v4.2.0

- Changing the behavior of table filters, allowing the use of relative periodic filters
- Changing the behavior of table filters, allowing it to be possible to copy filters without them being applied
- Add more information to event history details screen
- Fix event history list to support transactions sibs and sibs_v2
- Changing the behavior of opening details screen in transactions,event history lists
- Fix dropdown filter with multiple values selected (UI bug)
- Add share links on devices and logs lists
- Fix duplicated tables requests
- Fix exclusive filters cleanup
- Fix label on device history search bar

Changelog - v4.1.0

- Add missing information to transaction details screen
- Change red asterisk to (1) superscript to avoid confusion with required field (table filters)
- Add password-requiments on change password screen
- Export log data as CSV
- Implement share link on transactions screen
- Implement share link transaction details
- Fix 404 error when click on button “back” in event history pages
- Fix status toggle on operations list

Changelog - v4.0.1

- Fix navigation to back in event history pages

Changelog - v4.0.0

- Add event history page
- Add error logs page
- Change refresh data on dashboard page to use signalr
- Add support to exact match legend on table filter
- Transaction details (modal), show button to download proof
- Fix change language on your account screen
- Fix all date formatting according to preferred language

Changelog - v3.1.2

- Fix error on create app, inputs validation

Changelog - v3.1.1

- Fix renew session flow
- Add transaction proof download
- Add transaction proof validator
- Add support to marketing name on devices list

Changelog - v3.1.0

- Fix edit roles when role is locked
- Add support to secundary endpoint
- Add support to iPadOS on list of devices and dashboard

Changelog - v3.0.3

- Show transaction_id in transaction details modal
- Dashboard - Separating digits in graphics
- Fix permission validation to see dashboard page

Changelog - v3.0.2

- Update pipelines

Changelog - v3.0.1

- Fix fallback image on application selector

Changelog - v3.0.0

- Fix rule template condition modal when is closed to allow reopen
- Fix bug on transaction details risk_modules
- Replace functionalities endpoints to remove deprecated endpoints
- Add lock_out information on devices list

Changelog - v2.16.4

- Fix edit rule templates

Changelog - v2.16.3

- Fix operation presets page
- Fix device delete button in devices page

Changelog - v2.16.2

- Fix dropdown hover top bar while editing operation
- Fix dashboard tooltips
- Fix max results alert message on devices screen
- Fix operation parameter risk buttons: add rule,edit rule, delete rule

Changelog - v2.16.1

- Fix of max results warning message on devices list page
- Fix Dashboard transactions counter

Changelog - v2.16.0

- Add Dashboard PAGE
- Add RTAS back-end and front-end address in RTAS configuration pages
- Fix RTAS status toggle
- Fix bug on sidebar list when change preferred language

Changelog - v2.15.3

- Add pipeline id for e2e tests

Changelog - v2.15.2

- Add toogles to enable functionalities on create application screen
- Fix bug on edit application page to update the input values when inputs are changed multiple times

Changelog - v2.15.1

- Fix visual bug on create v2 application screen

Changelog - v2.15.0

- Add functionality access manager into settings -> functionalities screen
- Hide button of transaction details when transaction doesn't have details to show (pending)
- Add pophover on pagination on "…" button to choose page number to navigate
- Update internal dependecies
- Add suggested username on registration screen
- Fix some navigation inside admin area
- Fix limit/offset on API token applications list

Changelog - v2.14.2

Changelog - v2.14.1

- Fix administration roles screen navigation
- Fix oidc login/registration screens

Changelog - v2.14.0

- Add toogle to enable/disable email notifications
- Fix bug in pagination when changing the current page
- Fix broken URL path on settings, functionalities pages
- Fix validation of user permissions to access the settings page
- Change some labels on SIBS3DS V2 transaction details
- Fix API Tokens creation with multiple applications

Changelog - v2.13.1

- Fix callback timeout retro-compatibility on application settings

Changelog - v2.13.0

- Fix renew session flow
- Fix bug in pagination when changing the items per page in tables
- Change callback timeout on application settings to 1…30 seconds
- Remove unnecessary requests on operations page
- Fix requests when user switches from one app to another

Changelog - v2.12.0

- Add Transaction ID filter to transaction list
- Add security-related HTTP response headers

Changelog - v2.11.1

- Improve labels and translations
- Add action field to simulator on generic transactions screen
- Pagination is now limited to 100 requests, users must adjust their filters to get full results
- Fix bug in setting strict mode when editing money-v2 rules in operations
- Optimize API requests

Changelog - v2.11.0

- Improve labels and translations
- Add Realtime Authentication functionality
- Add new column and filter "Association date" into devices list

Changelog - v2.10.0

- Moved transaction duration in operations to the basic settings screen
- Added operations step names
- Small improvements on labels and tooltips
- Fix pagination bug (returning to page 1 in a table)

Endpoint Access

The TrustFactor Backoffice is deployed behind a firewall and is only available when accessed from certain IP addresses. To gain access to the Backoffice, you must first contact SecuritySide and request that your egress IP address(es) be added to the firewall allow list.

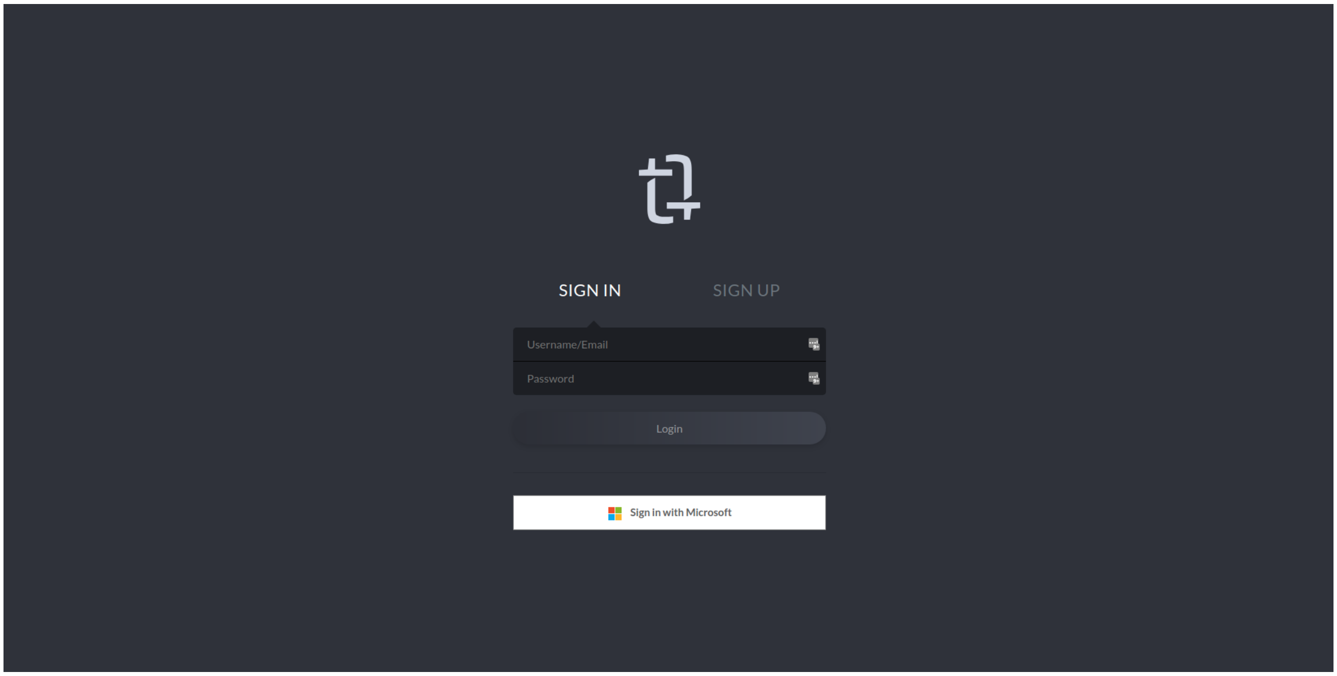
When the firewall blocks access, you will see a white page with just the word “Error” displayed on it. If this is happening then either:

1. Your egress IP address has not been added to the firewall. Use a website like ifconfig.io to figure out what your IP address is and send it over to SecuritySide Support.
2. Your egress IP address has been added but you are not using it for some reason. This is a common scenario when using VPNs, so please check your VPN connectivity if this is happening.

Authentication

Authentication in the TrustFactor BackOffice can be set up to work from two sources – local and OpenId Connect.

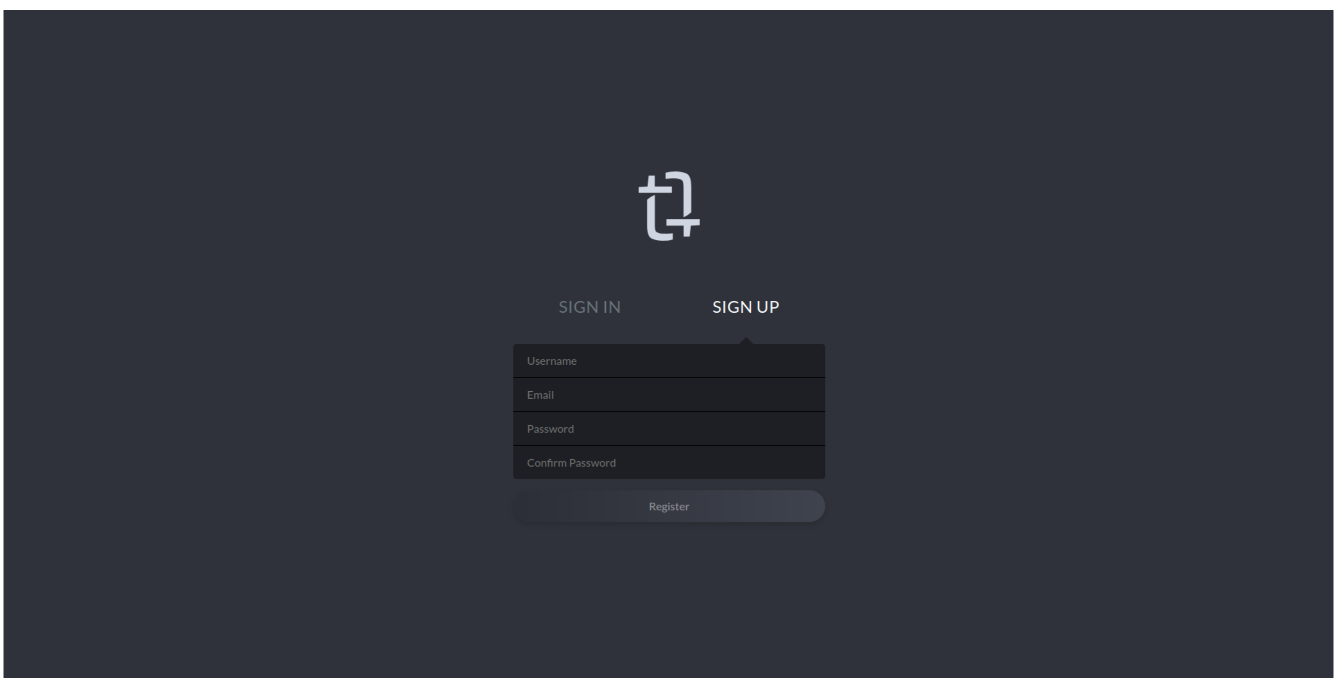
Local Authentication



When you enter the backoffice landing page, you are prompted to sign in. You can use either your email address or your username.

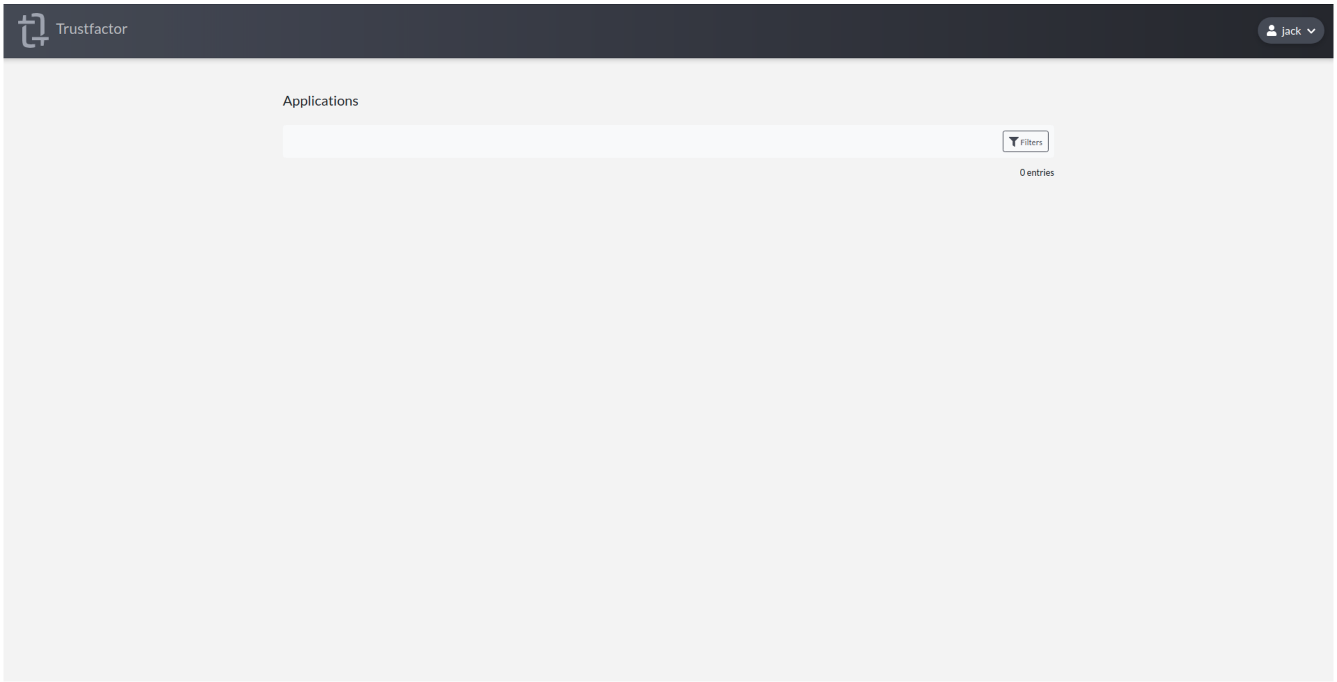
If you are already signed in you will be redirected to the app selector page.

User Sign Up



If you do not have an account, click "Sign Up" and fill out the registration form.

Registering an account does not give you privileges of any kind inside the backoffice, so it is normal to see an empty screen like the one below right after you register.



If the application you want to access already exists, you must ask an existing Application Owner for access. If the application needs to be created, you must contact SecuritySide Support to create it for you. SecuritySide can create an application for you with placeholder values that you can then change on the Application Settings or you can look through that section to find the required fields and provide them to SecuritySide when you request your new application. If you have a custom app deployment, then you may be able to create new applications yourself. See the *creating a new application section*.

OpenId Connect - CustomApps only

If you prefer to have SSO, you may request that your backoffice be linked to an OpenID Connect-capable provider. AzureAD is one of the supported providers, and you may enable access to the backoffice only to a specific AzureAD group. Please contact SecuritySide Support in order to set this up.

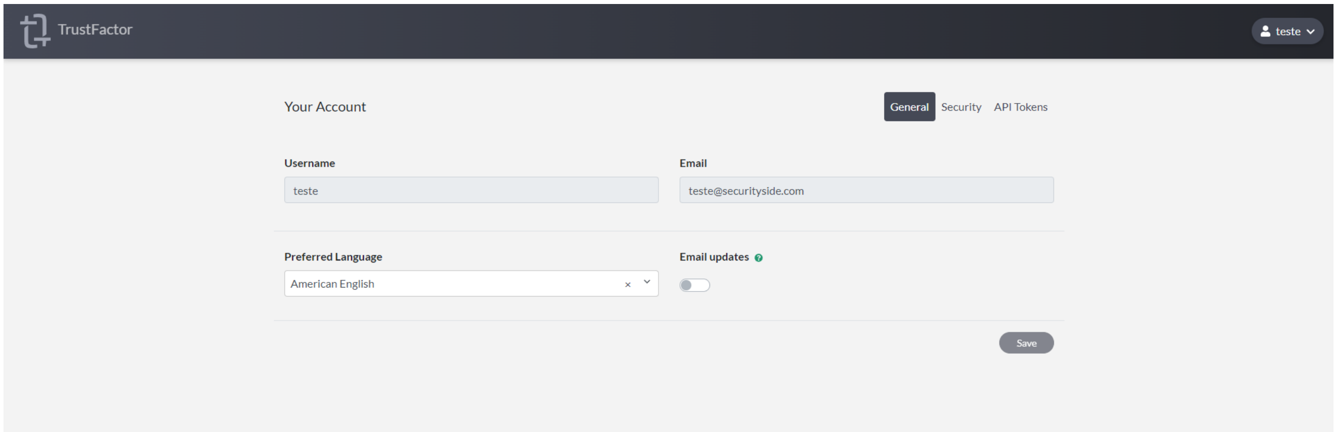
After your OpenId Connect has been configured, you may log in simply by pressing the "Login with Microsft" button on the login page. You may also request to remove local authentication in order to ensure users are signing in with Microsoft.

Your Account

After you have authenticated, you can press the top-right corner drop-down button with your username on it to reveal the *Your Account* option. This is where you can make changes to your backoffice account profile.

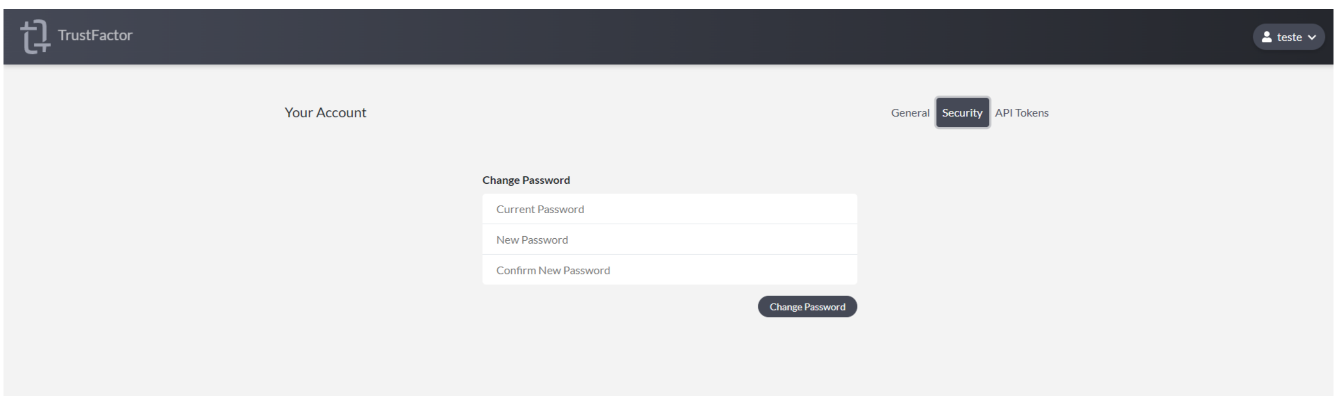
General

In the *General* panel, users can change their preferred language and also set whether to receive email updates.

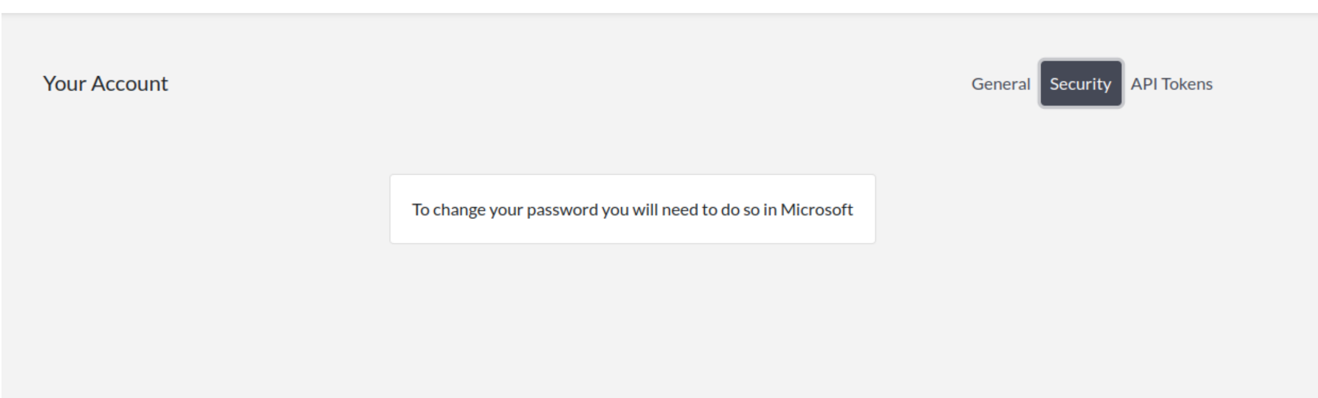


Security

In the *Security* panel, users can change their account password.



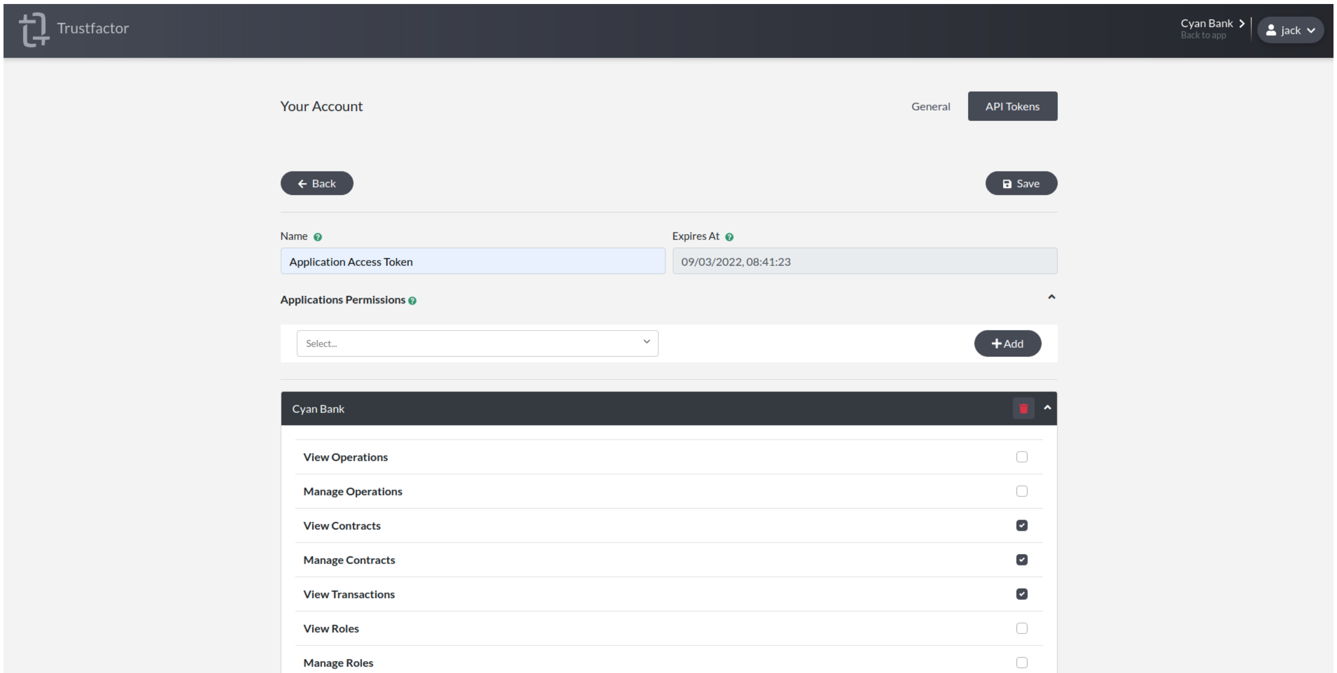
NOTE: Users that have signed up through OIDC cannot change their password here and must use their Identity Provider's page to do so.



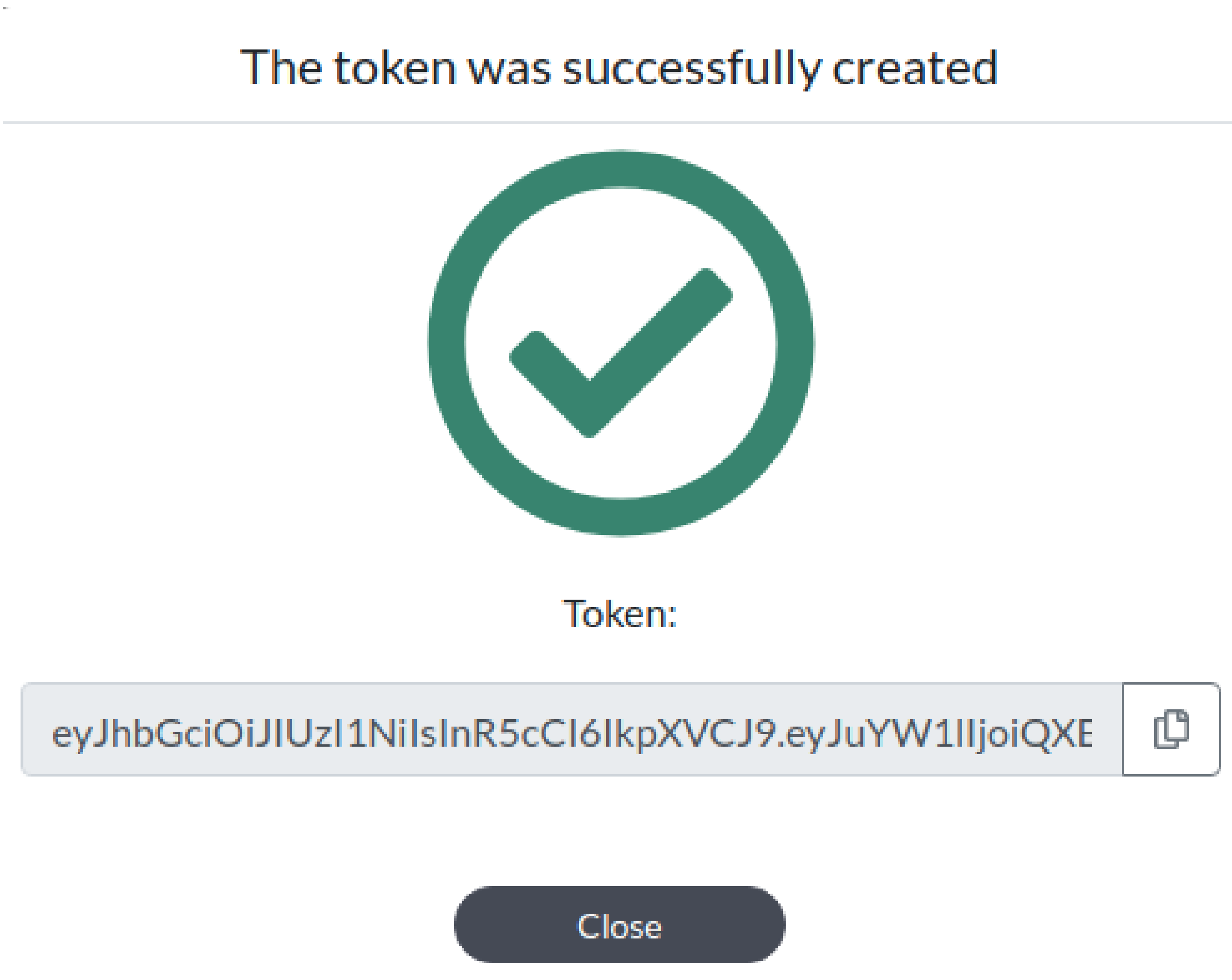
API Tokens

The backoffice back-end serves a REST API that can be used by other applications to perform actions within the context of one or more applications.

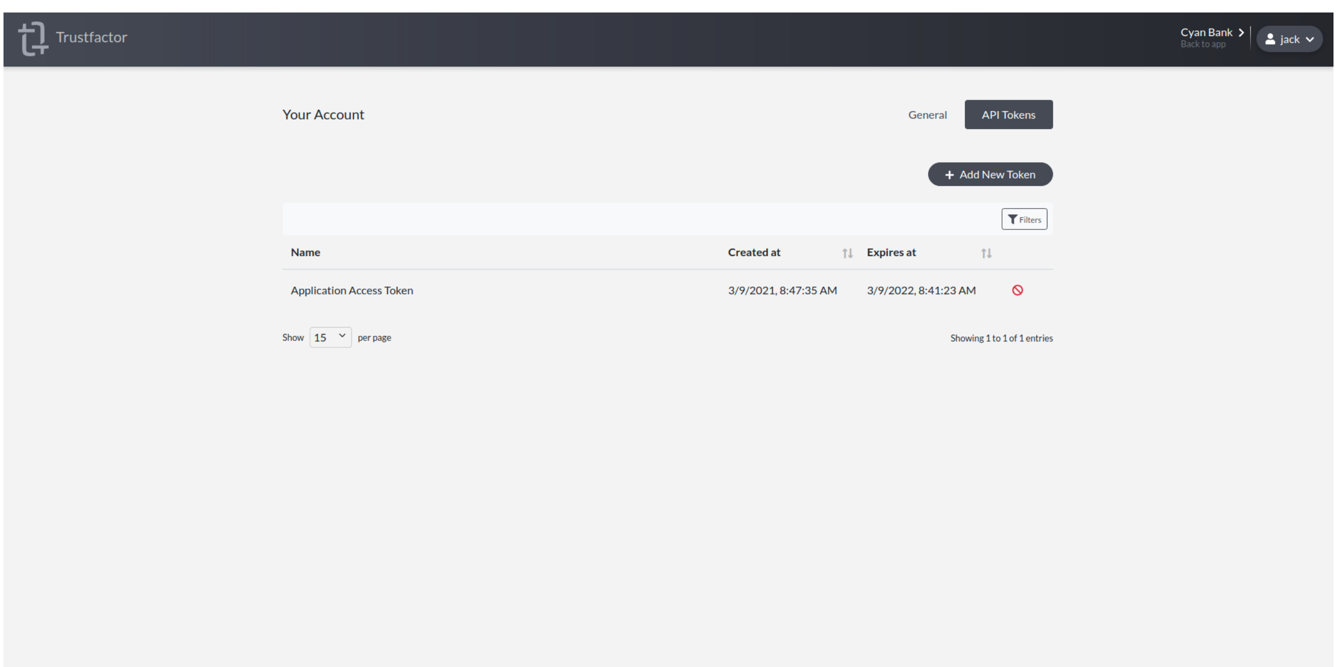
This section allows you to create API Tokens that can be used to authenticate against the back-end API. You can configure specific permissions for each application the Token has access to. See the Roles section under Applications for more information on application-level permissions.



When you are done configuring the API token's permissions, press the *Save* button and the token will be created and shown to you **once**. Save your token now as it will not be shown again.

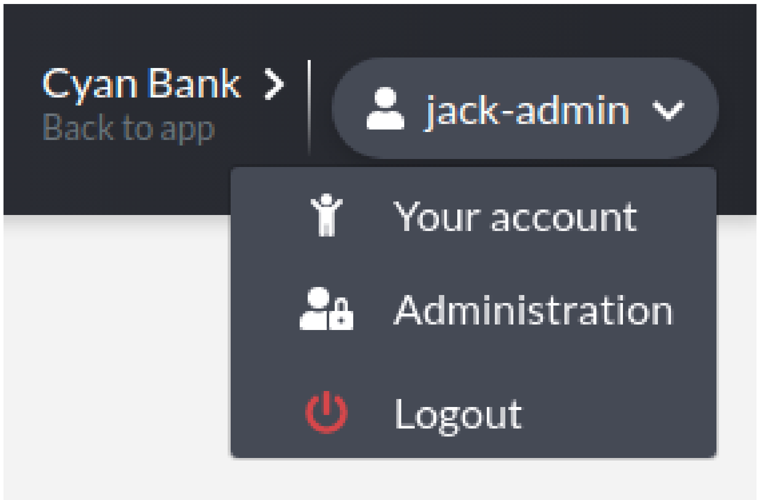


You can then manage (list or revoke) the tokens associated with your account in the main API Tokens view.



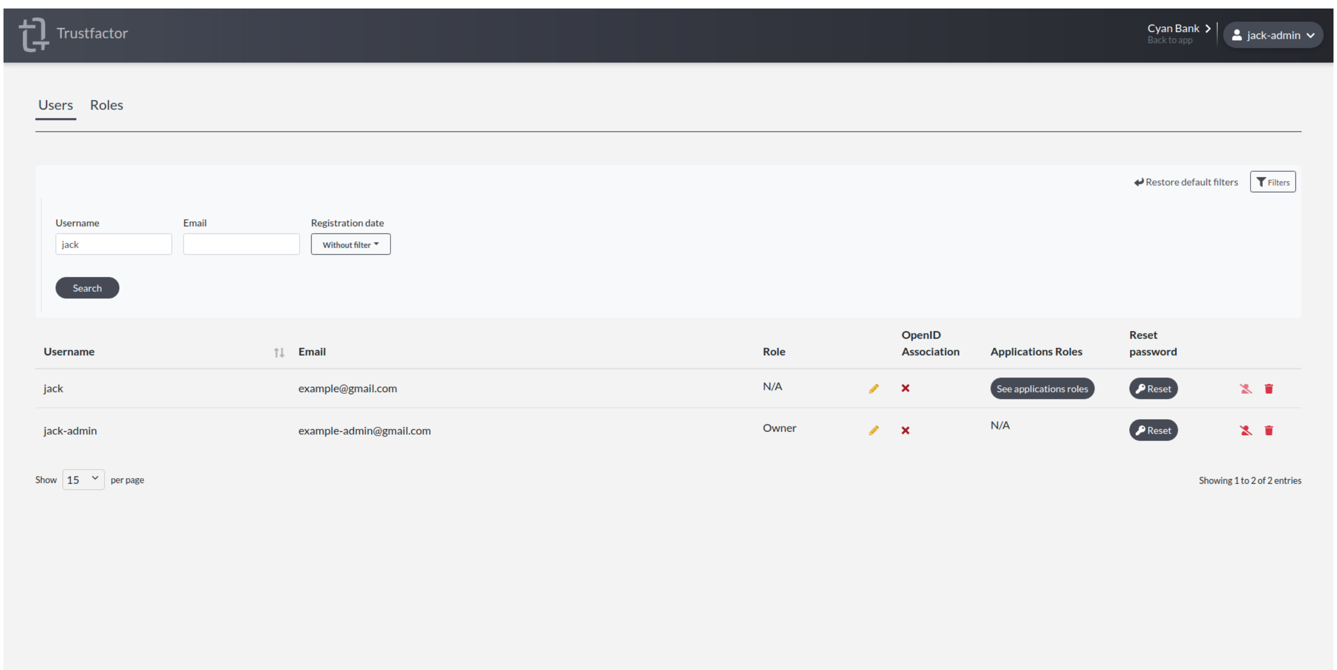
Administration - CustomApps Only

The TrustFactor backoffice has application-level permissions and roles and backoffice-level permissions and roles. This allows proper responsibility segregation between application owners and backoffice owners. If your user is a backoffice *owner*, you will see the *Administration* option in the drop-down menu with your username.



Users

In the **Users** tab, you can manage backoffice users.

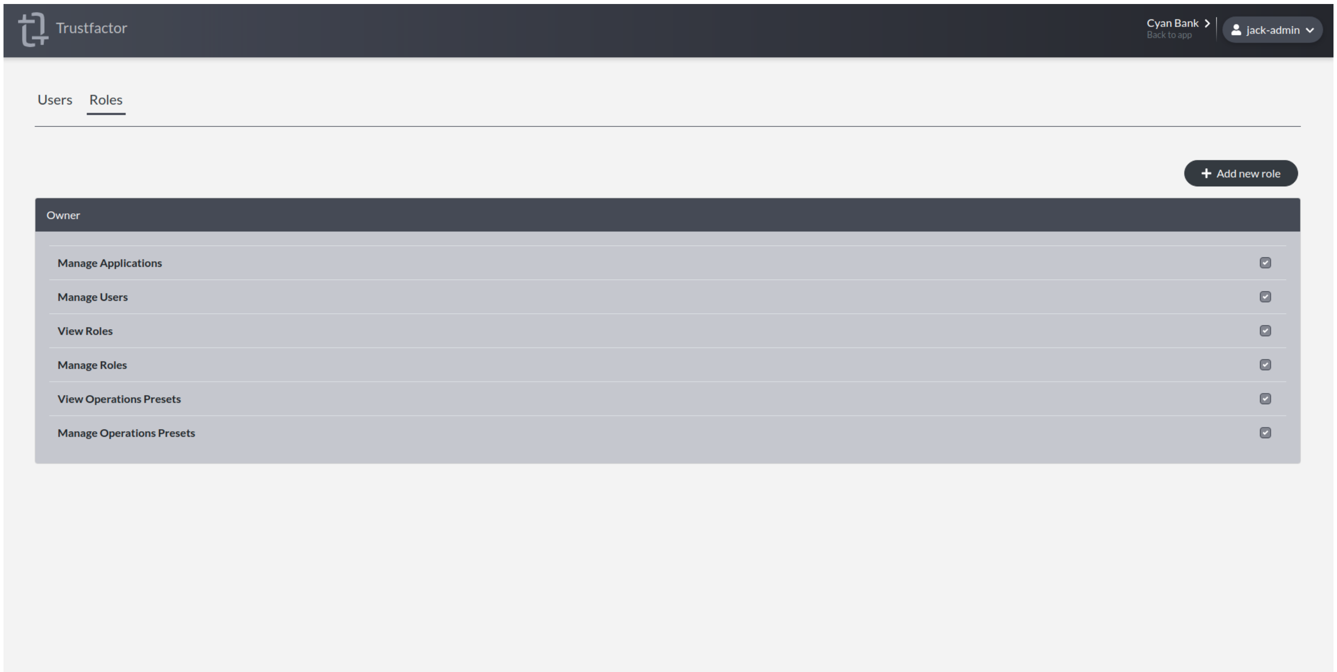


In this screen you can:

- Reset a user's password
- Update a user's backoffice role
- Check a user's OpenID Association
- See a user's application roles (applications in which they have application-level permissions)
- Delete a user

Roles

If you switch to the **Roles** tab, you can see the default role **Owner** has all of the permissions available at the backoffice-level. You can create new roles using the different permissions available suitable for your needs.

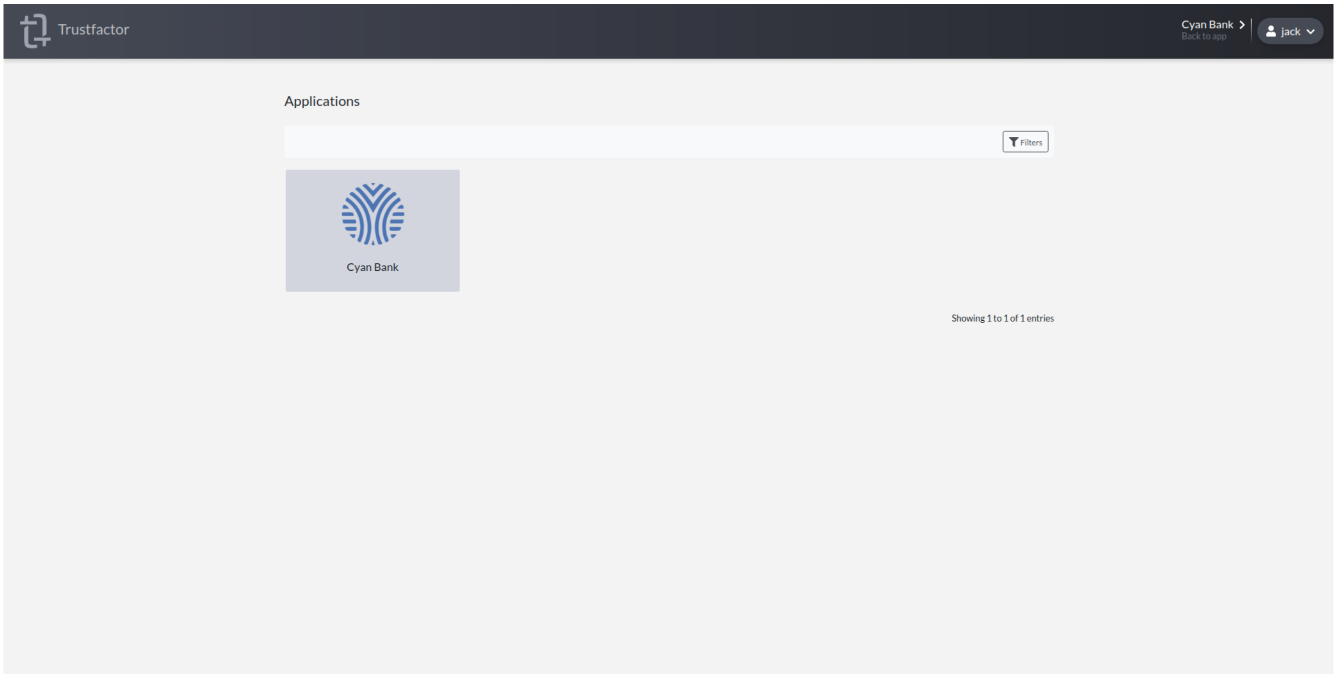


Backoffice-level Permissions

Below is a list of backoffice-level permissions:

- **Manage Applications** - Backoffice users with "Manage Applications" can create, delete and otherwise manage any application in the backoffice.
- **Manage Users** - Backoffice users with "Manage Users" can assign backoffice-level roles to backoffice users.
- **View Roles** - Backoffice users with "View Roles" can see the different roles available and what permissions they hold.
- **Manage Roles** - Backoffice users with "Manage Roles" can create new backoffice-level custom roles.
- **View Operations Presets** - Backoffice users with "View Operations Presets" can view and use operation presets in their applications.
- **Manage Operations Presets** - Backoffice users with "Manage Operations Presets" can create operation presets for all applications. # Applications

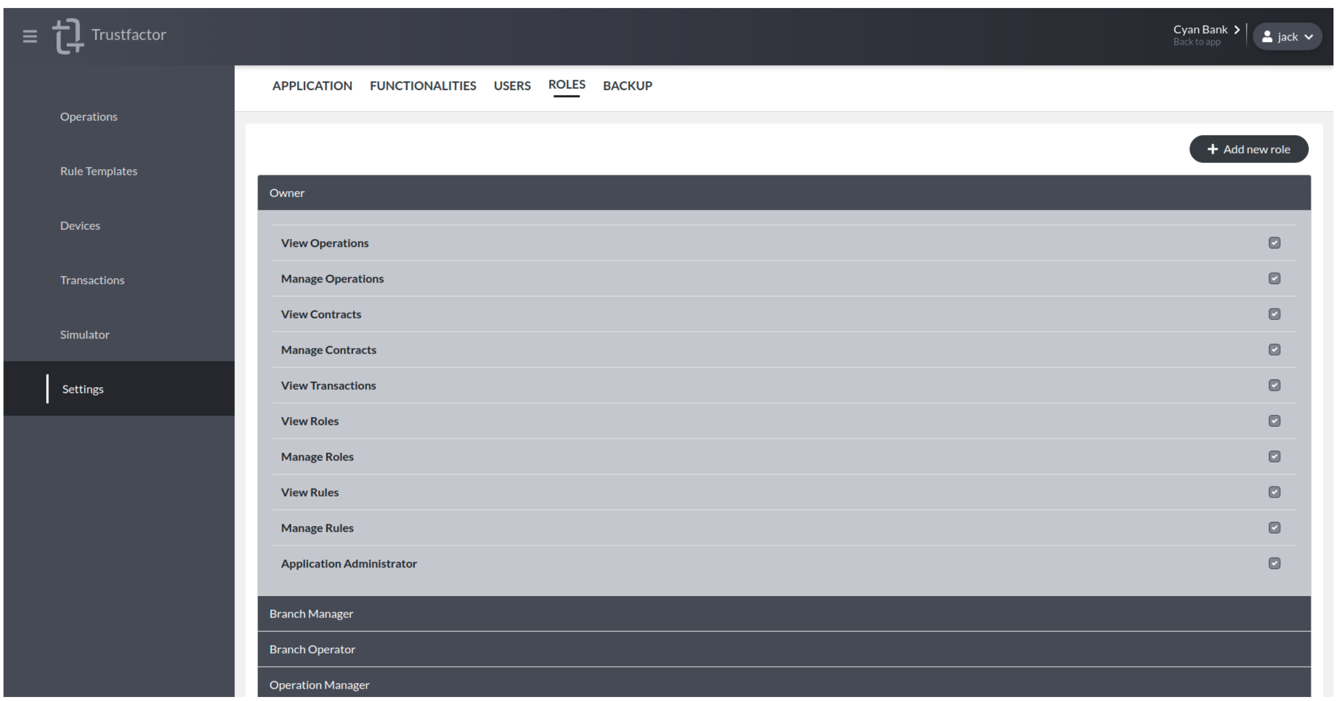
The first screen you will find upon login is the application selector. This is where you can navigate to different applications you have access to.



If you press the application card you will enter the application context on the default option, which is Devices.

Before getting into each of the functionalities provided in the application context, you should get familiar with application permissions and roles.

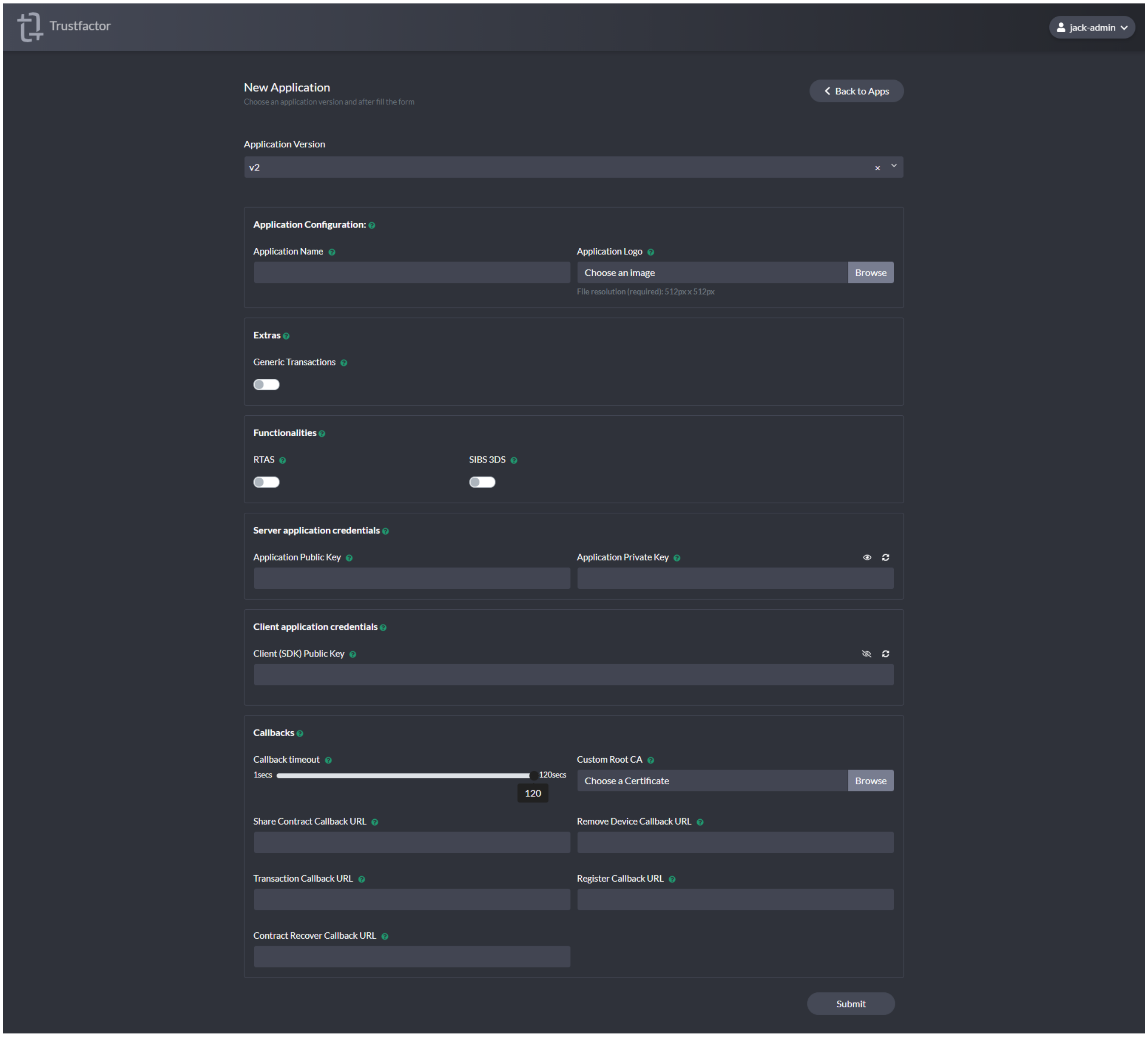
For more information about application permissions and roles, check out the *Roles section*.



Creating a new application - CustomApps Only

If you have a custom app deployment, you will be able to create new applications on the backoffice at will, provided your user has the “manage applications” backoffice-level permission.

In the application selector, you will see a *+ Add Application* button which will take you to the new application screen.



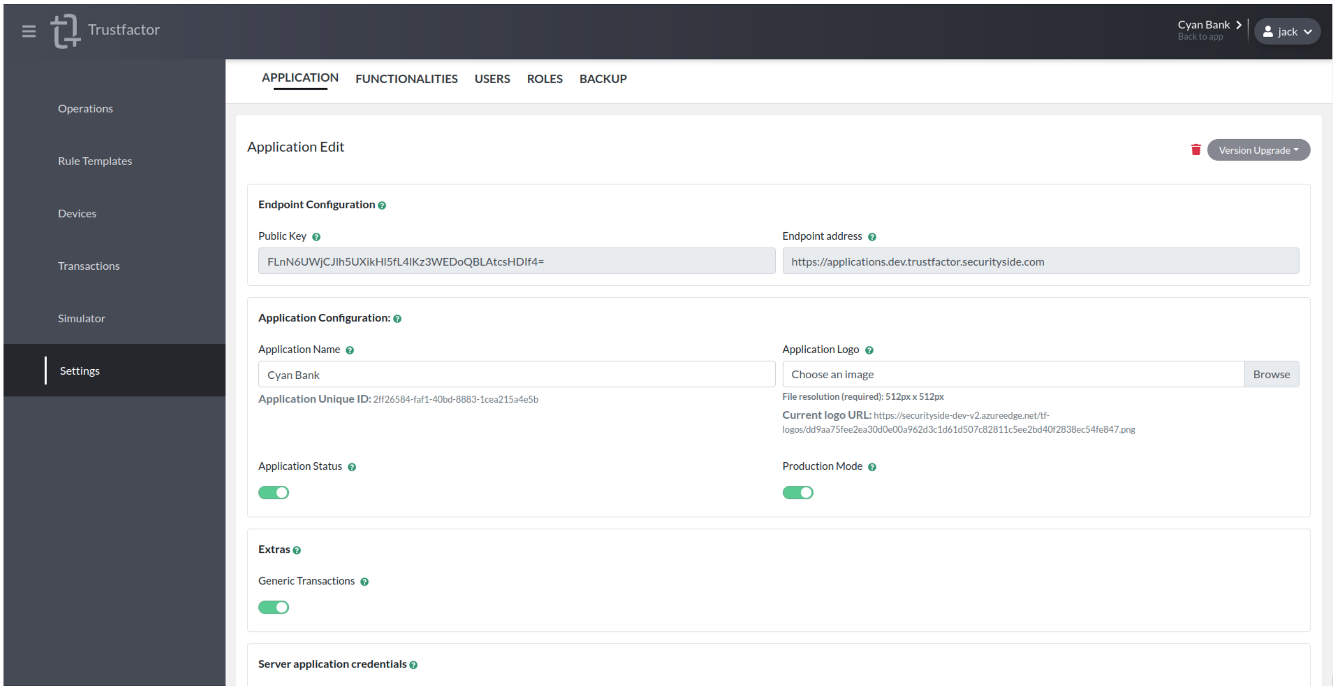
To create a new application, you need to fill out the fields in the *New Application* screen. These fields are the same fields used in the Application Settings menu.

Settings

Application

Permissions required to view this screen: - **Application Administrator**

Permissions required to make changes in this screen: - **Application Administrator**



Endpoint Configuration In the application settings is where application owners can change and view configurations for their application.

The first section holds endpoint configurations. These should be used to initialize the TrustFactor SDK. The EndPoint Public Key is used to encrypt messages to the TrustFactor Application Endpoint. The Endpoint URL is also shown in this section.

Application Configuration In the Application Configuration section, Application Administrators can change some of the main configurations of their application.

- **Name**

The name of the application as it is shown in the TrustFactor App

- **Status**

Whether or not the app is enabled. If the app is disabled, the SDK methods *CreateRegisterCode*, *CreateTransactionV2*, *CreateTransactionV3* and *UpdateTransactionStatus* will be blocked from being used. This is useful if you are migrating users from one application to another and don't want to allow new users or transactions to be created in the disabled application.

- **Application Logo**

This is the app logo shown in the TrustFactor App when users receive a new transaction to authenticate. It must be 512x512 px.

- **Production mode**

Enabling production mode allows the TrustFactor App to abort functionality if the application does not respond with a 200 OK HTTP code in the callback. This prevents state from diverging between the application and the TrustFactor services and should be enabled for production applications. It is sometimes useful to disable Production mode during development and integration of TrustFactor with an application, because for example a given callback has not been implemented yet. If that is the case you can disable production mode here.

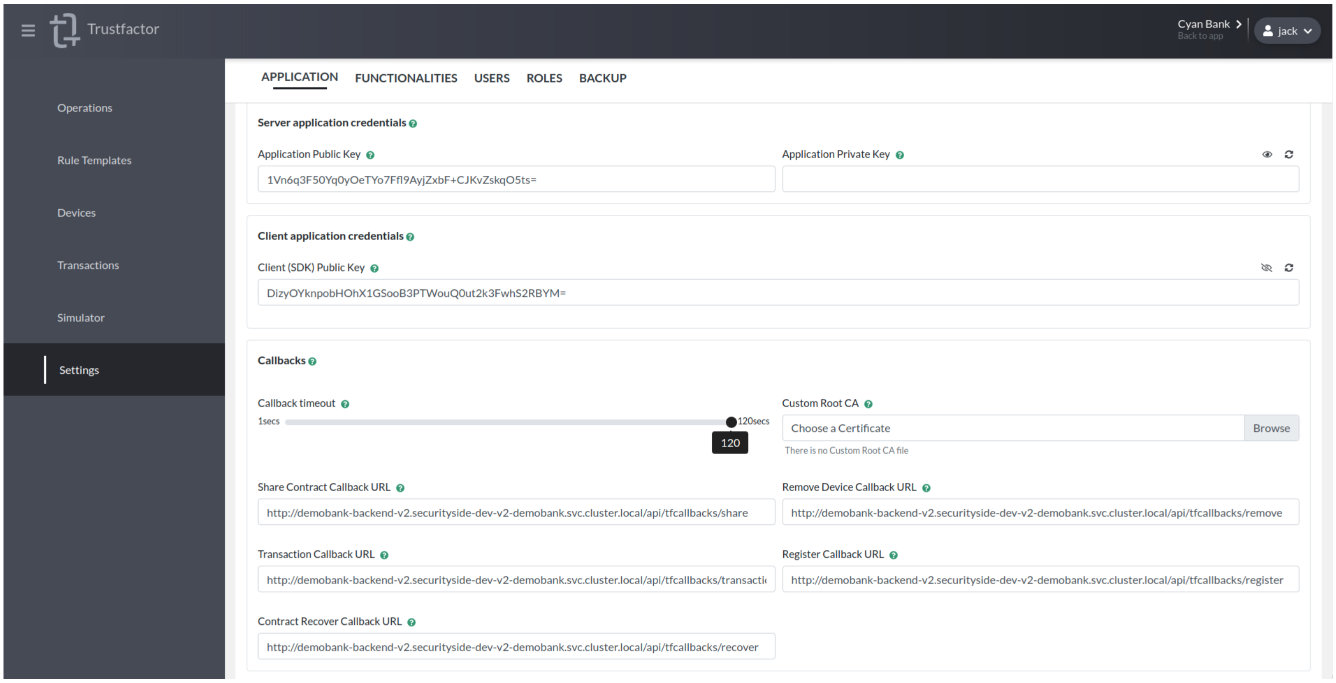
The affected functionalities are:

- Device Registrations
- Device Removals
- Contract Sharing

- Contract Recovery

Extras This section is used to enable extras on the TrustFactor SDK. A good example is Generic Transactions, which allow developers to generate authentication requests on the fly without having to configure an operation for them.

Toggling Generic Transactions on, enables this functionality on the SDK.



Server application credentials This section is where the authentication credentials for the server-side application can be configured and generated. Use the generate button to create Public / Private key-pair to register your TrustFactor App. Both keys will be sent to the server and then used to send and receive requests on behalf of your application.

Client application credentials This section is where the authentication credentials for the SDK application can be configured and generated. Use the generate button to create Public / Private key-pair to use with your TrustFactor SDK. Only the public key will be sent to the server here. The private key never leaves your browser and when you generate a new keypair you must save the private key to use with the TrustFactor SDK. Only SDKs configured with this key will be allowed to communicate with your application on the TrustFactor App, so without it, you will not be able to use the SDK.

Callbacks In the callback section you can configure:

- **Callback URLs**

The URLs at which your application is receiving the asynchronous callbacks sent from the TrustFactor cloud services for the relevant events.

- **Callback timeout**

The HTTP client timeout, seconds after which the TrustFactor services should close a connection to your application. Slower applications must configure the callback timeout here as they see fit. The current maximum timeout is 120 seconds.

- **Custom Root CA Certificate**

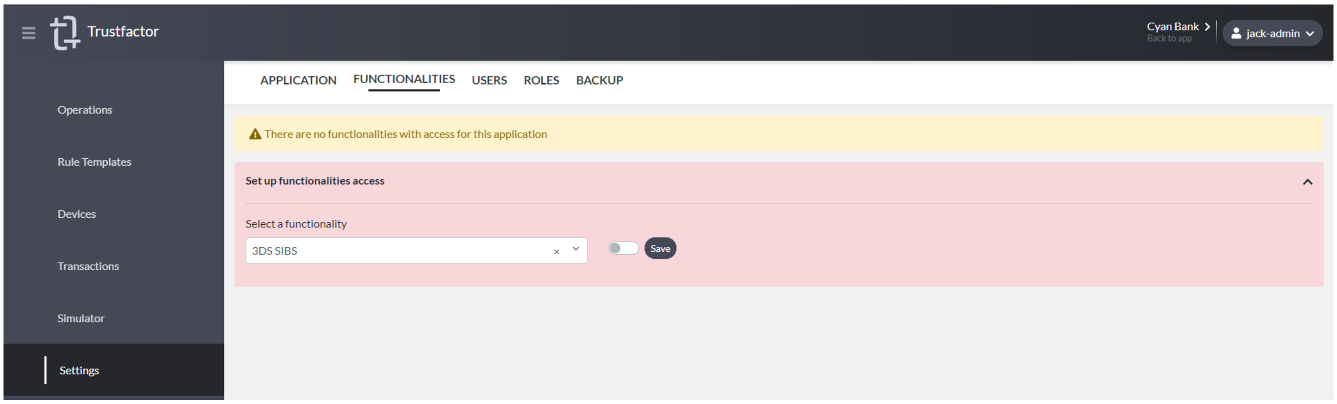
It is sometimes useful to have a self-signed certificate used for development or testing endpoints of your application. This option can be used to set the public certificate of the Root Certificate Authority that signs the TLS certficate on your application callback endpoints.

Functionalities

This section is used to configure functionalities or modules added to TrustFactor.

Set up functionalities access Permissions required to view this screen: - **Manage Applications**

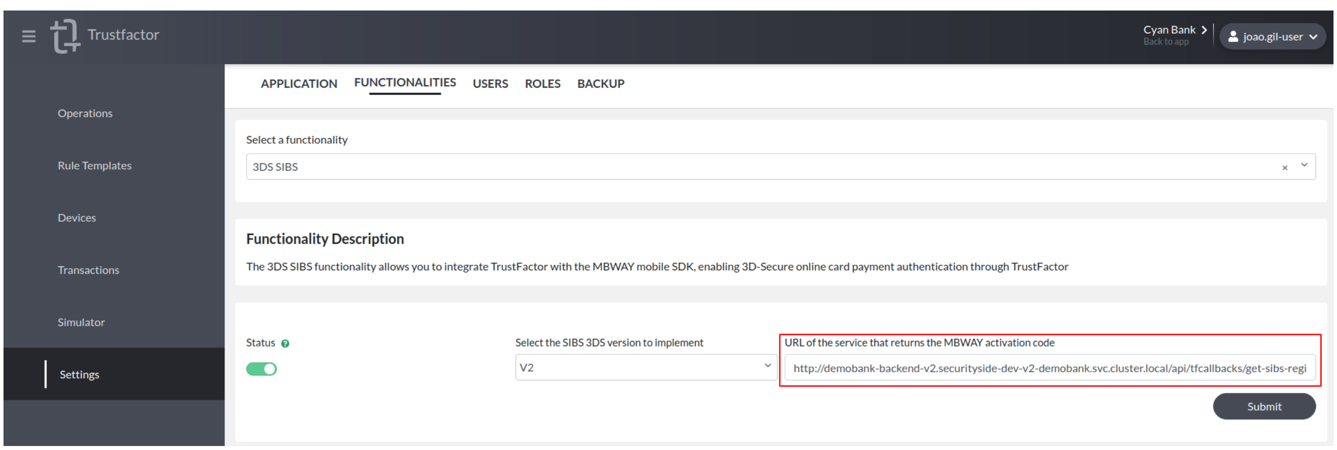
First, before configuring a feature, you need to give the application access to allow it to be used. This depends on the features that are active for the environment. ex: "3DS SIBS", "Real-Time Authentication Service".



After enabling access to at least one feature, you will be able to see the screens documented below and can proceed with the configuration of the respective features.

3DS SIBS - Custom Apps only The 3DS SIBS functionality allows you to integrate TrustFactor with the MB-WAY mobile SDK, enabling 3D-Secure online card payment authentication through TrustFactor. This functionality requires embedding the SIBS MBWAY SDK on the TrustFactor Agent applications and so is available for custom apps only.

There are two different versions of the SIBS MBWay implementation: **v1** and **v2**. **v1 is not recommended** for new implementations, as it lacks the ability to interact with pending authentication requests, knowing the final status of the authentication, among others, so new implementations should always target v2.



The **SIBS 3DS v2** implementation requires:

- C# SDK v3.1.0 or higher
- Java SDK v3.2.0 or higher

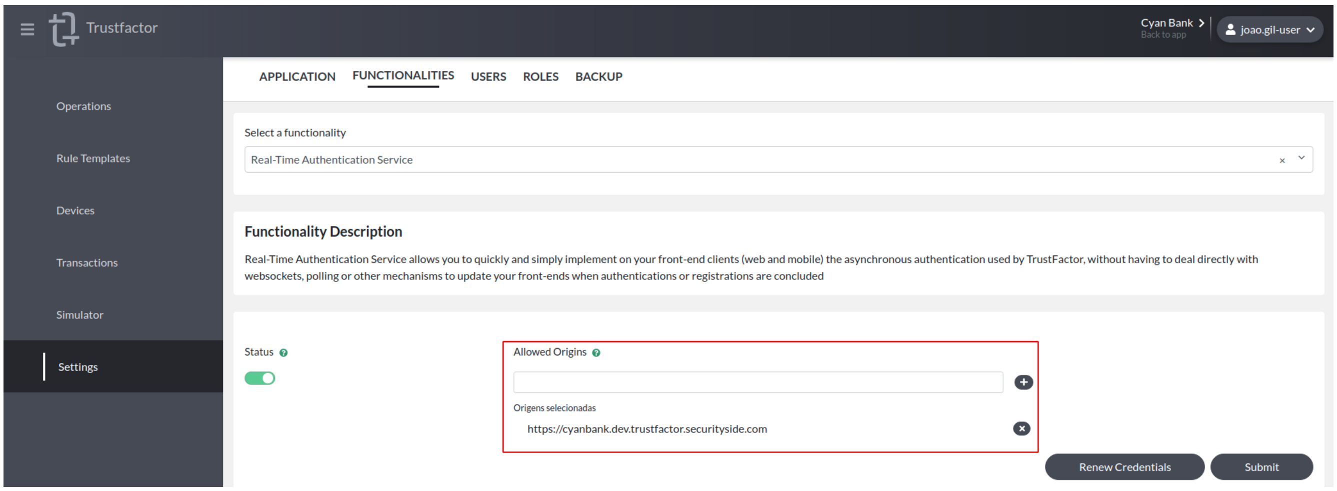
In these SDK versions there are methods to create SIBS 3DS V2 transactions, to update the status of SIBS 3DS V2 transactions when the decision callback is received from SIBS and also to get a new SIBS MBWAY SDK activation code. This last endpoint is required and must be configured in the screen as shown above.

Real-Time Authentication Service Real-Time Authentication Service (RTAS) allows you to quickly and simply implement on your front-end clients (web and mobile) the asynchronous authentication used by TrustFactor, without having to deal directly with websockets, polling or other mechanisms to update your front-ends when authentications or registrations are concluded.

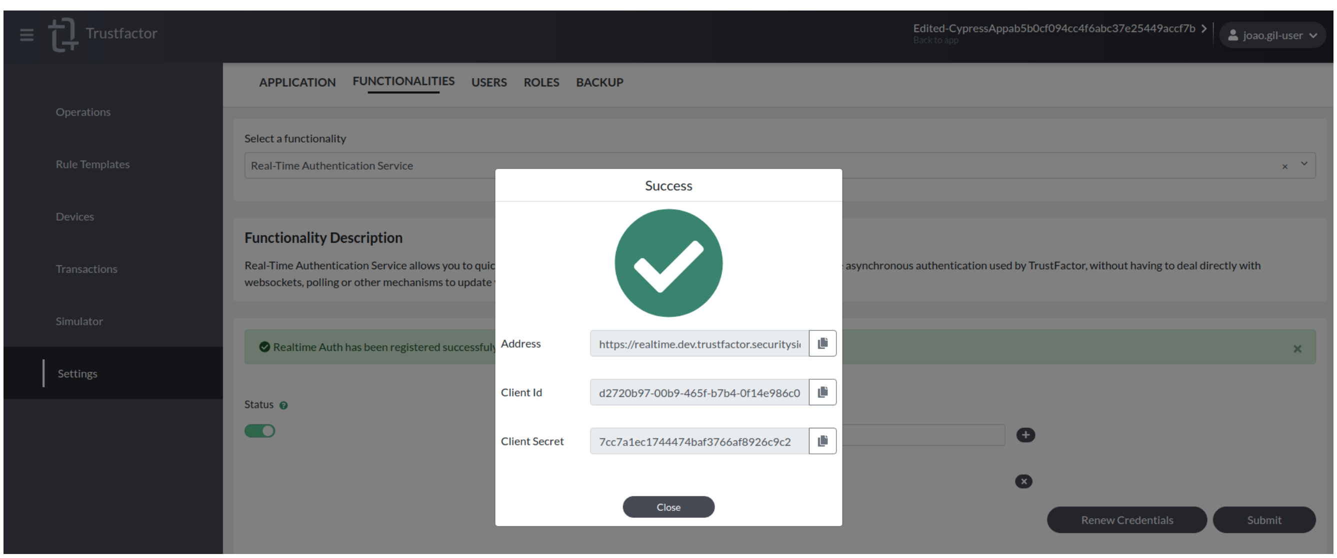
This external service uses its own SDK for backend applications and also has a Javascript component to use in the front-end application (mobile or web). Please contact SecuritySide support for more information on this service and how to use it.

In order to get credentials for RTAS, you can use the functionalities tab. You need to set:

- **Allowed Origins**: Define the front-end application origins (http://example.com) from which the Javascript code will contact RTAS. If you intend to use mobile webviews, don't forget to add *null* as well.



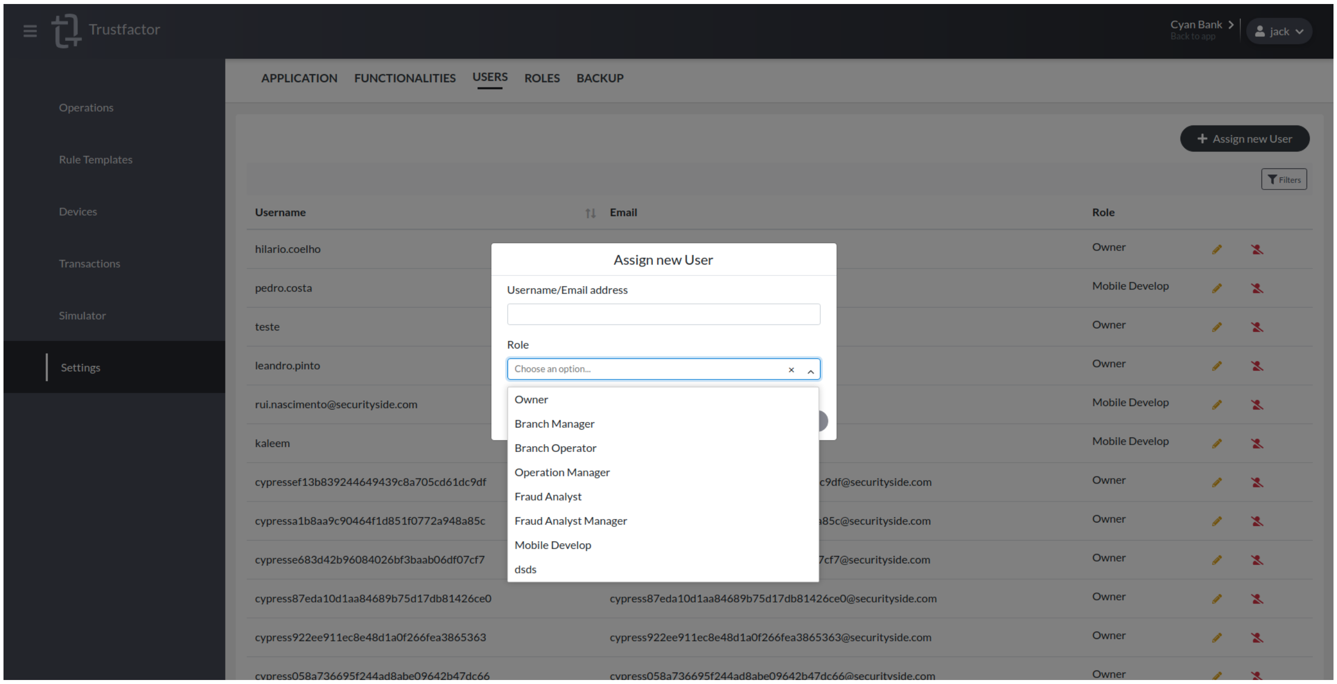
After activating the service, you will see a pop-up showing you the endpoint address, the clientId and clientSecret of your application, all of which will be needed on the RTAS backend and frontend SDKs in order to authenticate your users.



Users

Permissions required to view this screen: **- Application Administrator**

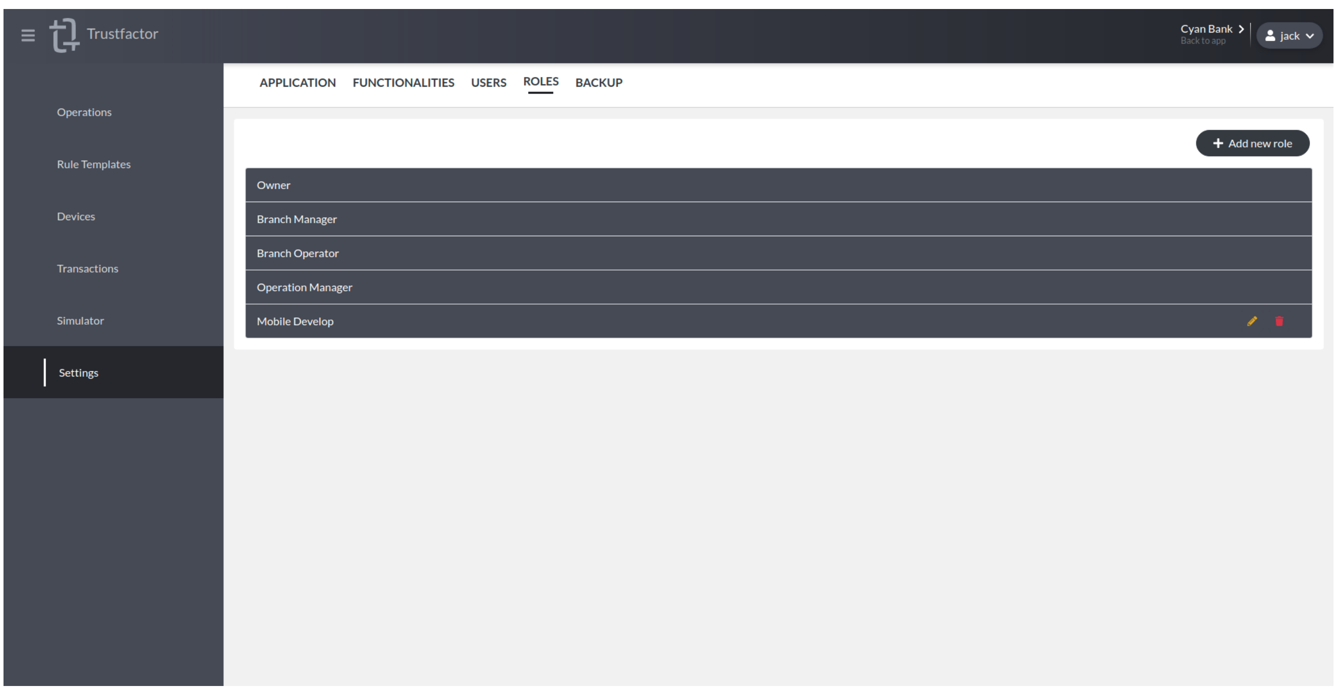
Permissions required to make changes in this screen: **- Application Administrator**



In this section you can assign users a role on your application. Access to this screen requires “Manage Roles” permission.

By pressing the “Assign new user” button, you can give any registered backoffice user access to your application. You can use their email address or username and pick a role from the dropdown menu.

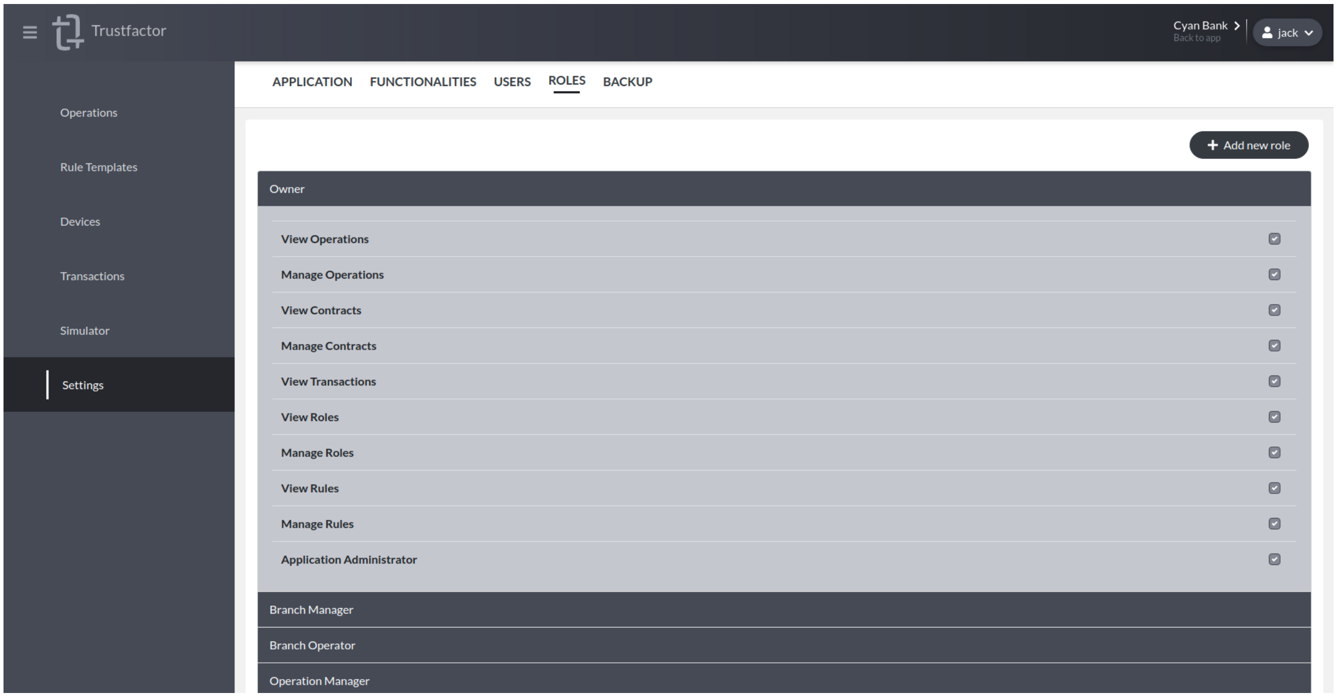
Roles



Permissions required to view this screen: - **View Roles**

Permissions required to make changes in this screen: - **Manage Roles**

Roles is where you can create or view user roles in your application. For backoffice applications, there are several application-level user permissions. These allow users to access functionality to read data or even change data using write permissions. All of these settings are configured in the Settings menu. Permissions can be aggregated into user-defined Roles and then Users can be assigned these Roles.



Below is a list of all application-level permissions:

- **View Operations** - Read-only access to the Operations tab
- **Manage Operations** - Write access to the Operations tab (adds View Operations automatically)
- **View Contracts** - Read-only access to the Devices tab
- **Manage Contracts** - Write access to the Devices tab (adds View Contracts automatically)
- **View Transactions** - Read access to the Transactions tab
- **View Roles** - Read-only access to the Roles tab in the settings
- **Manage Roles** - Write access to the Roles tab in the settings (adds View Roles automatically)
- **View Rules** - Read-only access to the Rule Templates tab
- **Manage Rules** - Write access to the Rule Templates tab (adds View Rules automatically)
- **Application Administrator** - Application Administrator allows access to the settings tab and assign back-office users their roles

There are a few roles already built-in with every application:

- **Owner**

The owner role has all permissions, including Application Administrator, and therefore can access all functionalities and configurations on the application. There must always be at least one Application Administrator at any time for an application.

- **Branch Manager**

This role was designed with front-office or contact center users in mind. They only have "manage contracts" permission which allows them to remove devices from users who get in touch with your support team. They cannot view the users' transaction history or any other information on the application.

- **Branch Operator**

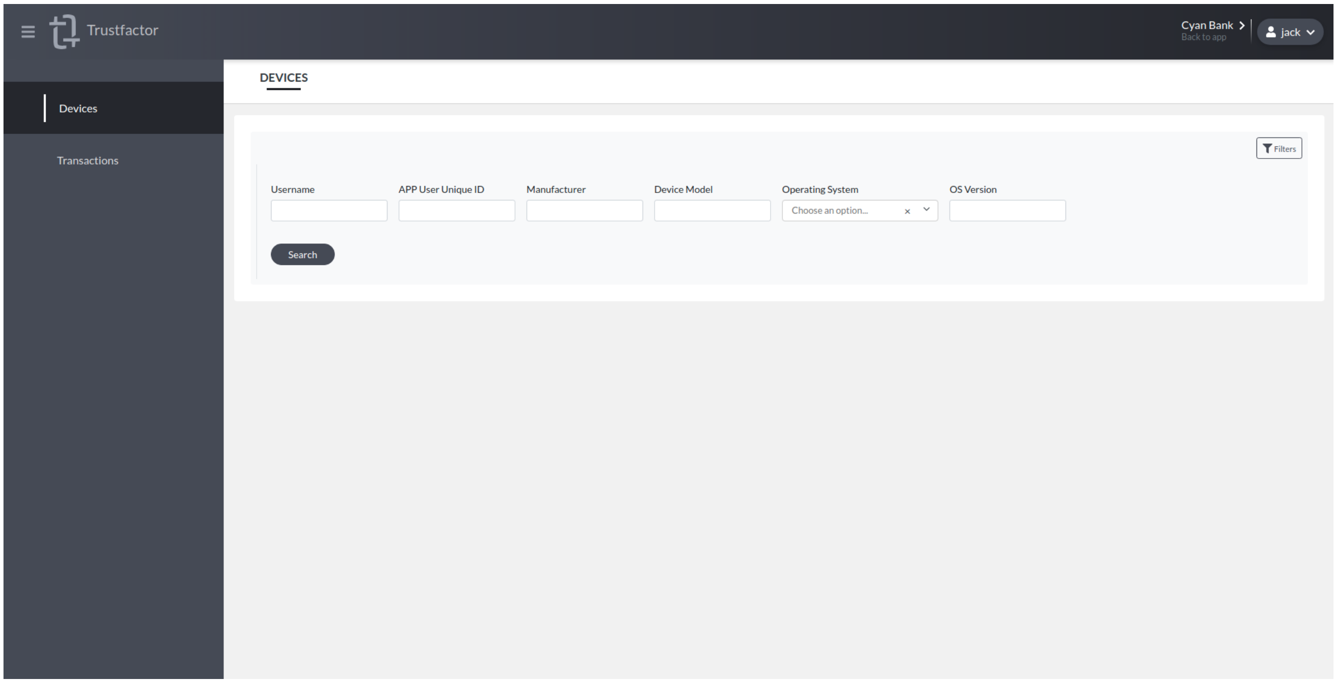
This role can only check if a user has a device enrolled with the application, but cannot remove it. It has no other permissions.

- **Operation Manager**

This role was designed for a user that manages the operations and risk rules. This user has read/write access to the created operations in the application and risk rule templates. They can change risk rule values for live operations, create new operations or edit live ones.

Restricted Application Access

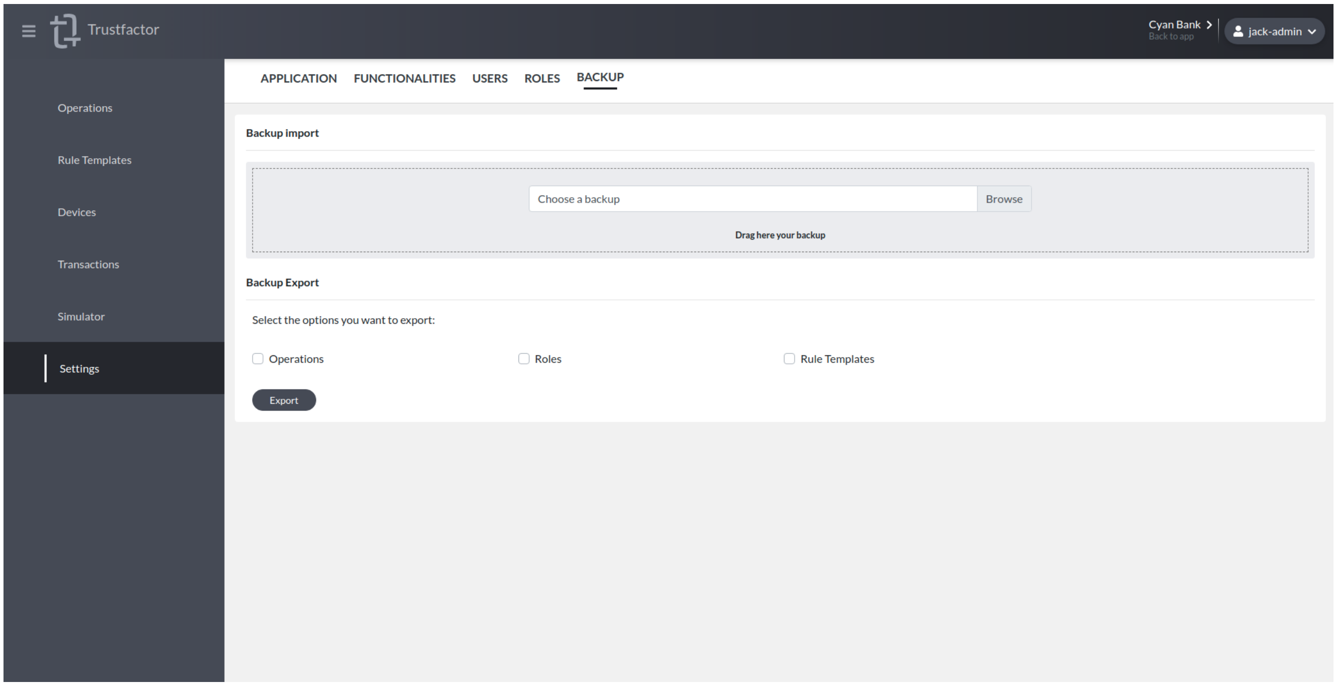
Users who have restricted permissions will not see all of the application menu tabs. For example, if a user has a custom role that only has the permissions *Manage Contracts*, *View Contracts* and *View Transactions* and they click the application card on the application selector menu, they will have the view as seen in the screenshot below with only two menu tabs shown.



Backup

Permissions required to view this screen: - **Application Administrator**

Permissions required to make changes in this screen: - **Application Administrator**



The backup tab is used to export and import different settings in the application. This allows users to store backups of configurations and restore them easily into new applications for different environments.

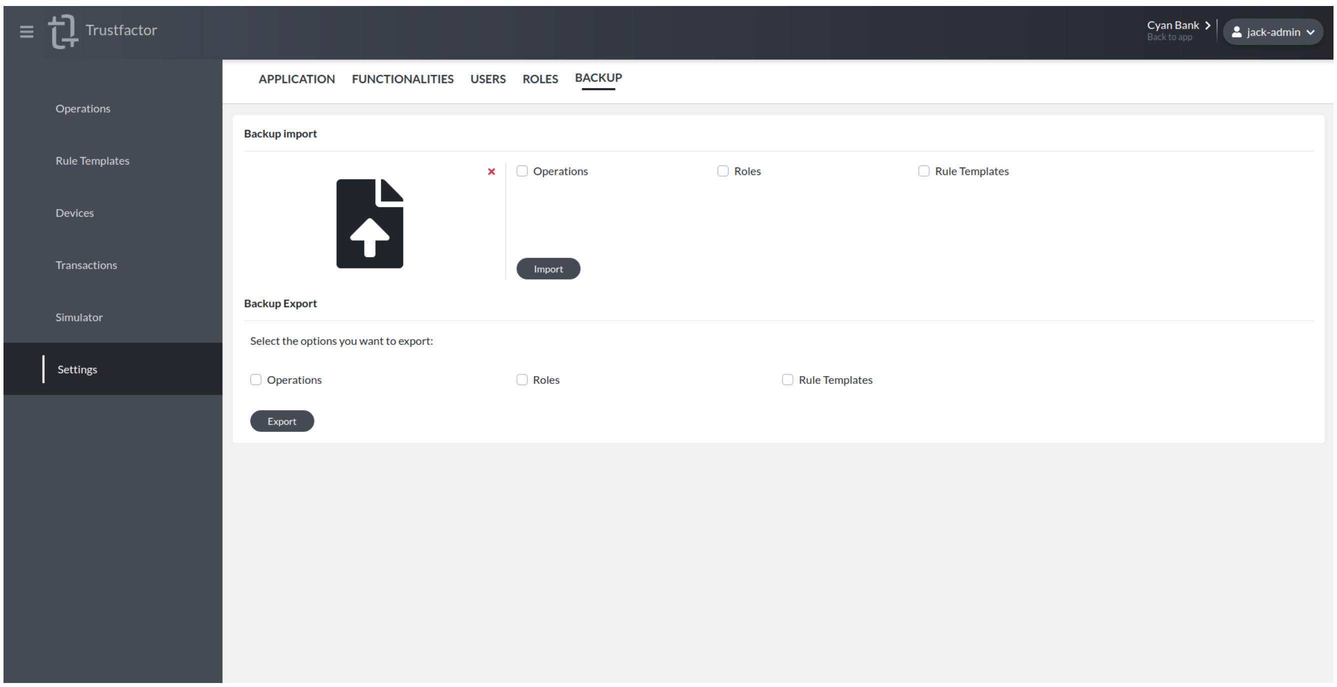
You can back up and restore:

- Operations
- Custom Roles
- Rule Templates

Restore To restore, choose a backup file on the Browse button in the "Backup import" section.

After the file is loaded, you can then check which options you want to import from that file and press the "Import" button to proceed with the import.

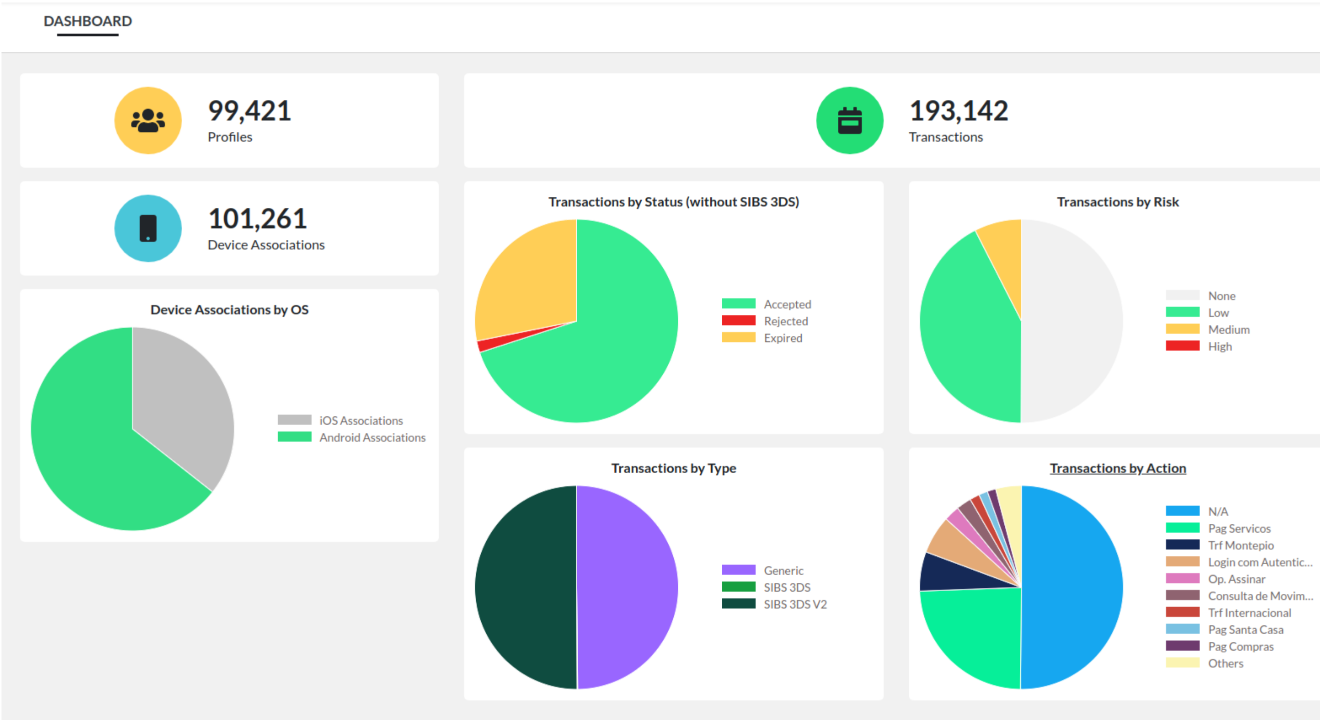
NOTE: Restoring operations will overwrite existing configurations. If, for example, you already have an operation with action "login" and restore a backup that also has an operation with action "login", the operation defined in the backup will override the existing configuration.



Backup To back up, simply tick off the options you want to export and press the export button. This will output a JSON file with the configurations that can be used in the *restore screen*. ## Dashboard

The dashboard can be used to get real-time statistics of TrustFactor use on your application. Here we show you data about enrolled devices, profiles and the transaction usage and risk.

Permissions required to view this screen: - **Application Administrator**

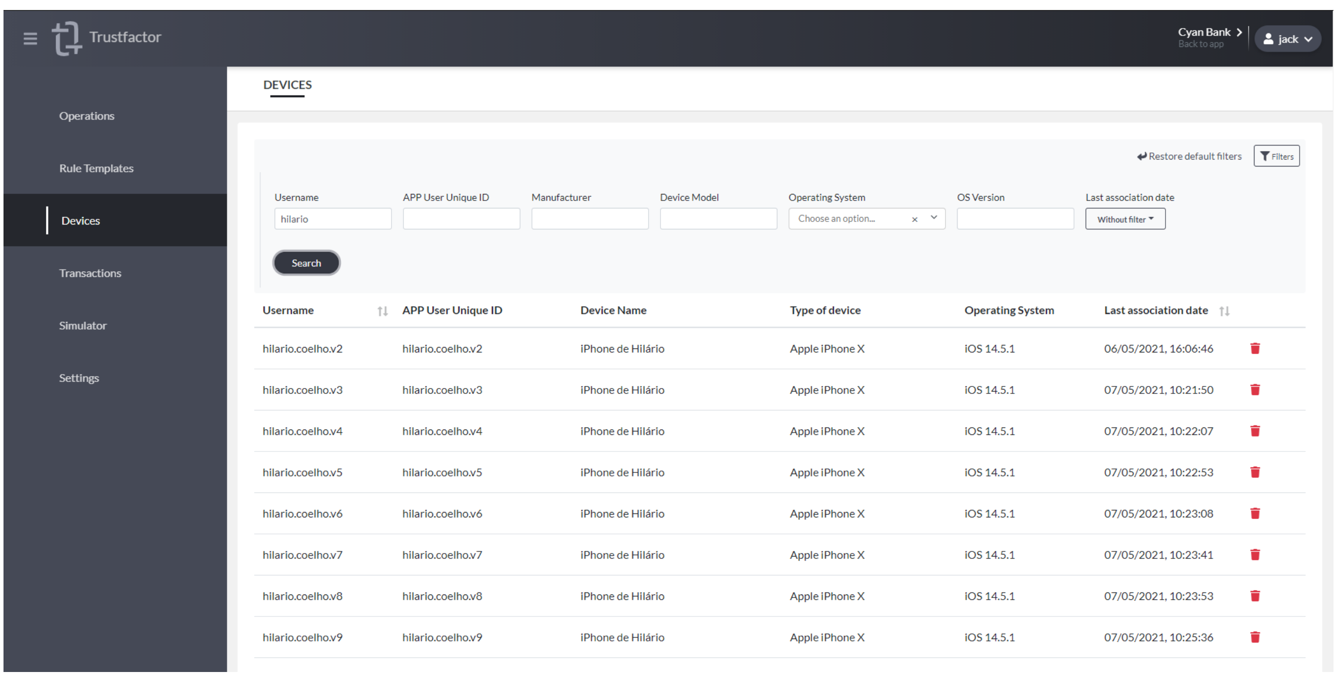


Devices

In the devices screen you can search for user devices associated with your application. You can also remove them if the user has lost access to the device and needs to re-enroll their TrustFactor App.

Permissions required to view this screen: - **View Contracts**

Permissions required to make changes in this screen: - **Manage Contracts**



By default, the search results are hidden and you must select a filter to search.

You can filter for:

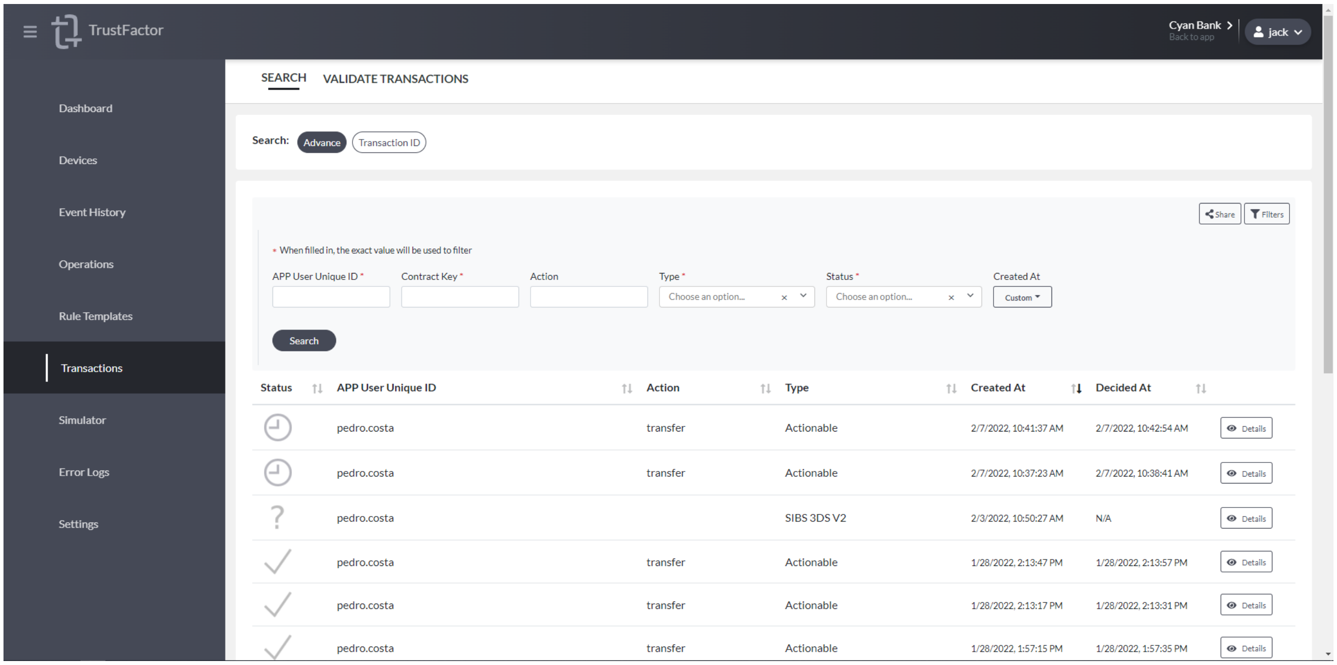
- Username
- App User Unique ID

- Device Manufacturer
- Device Model
- Device Operating System
- Device Operating System Version
- Association date

Transactions

Transactions Search - Advanced

Permissions required to view this screen: - **View Transactions**

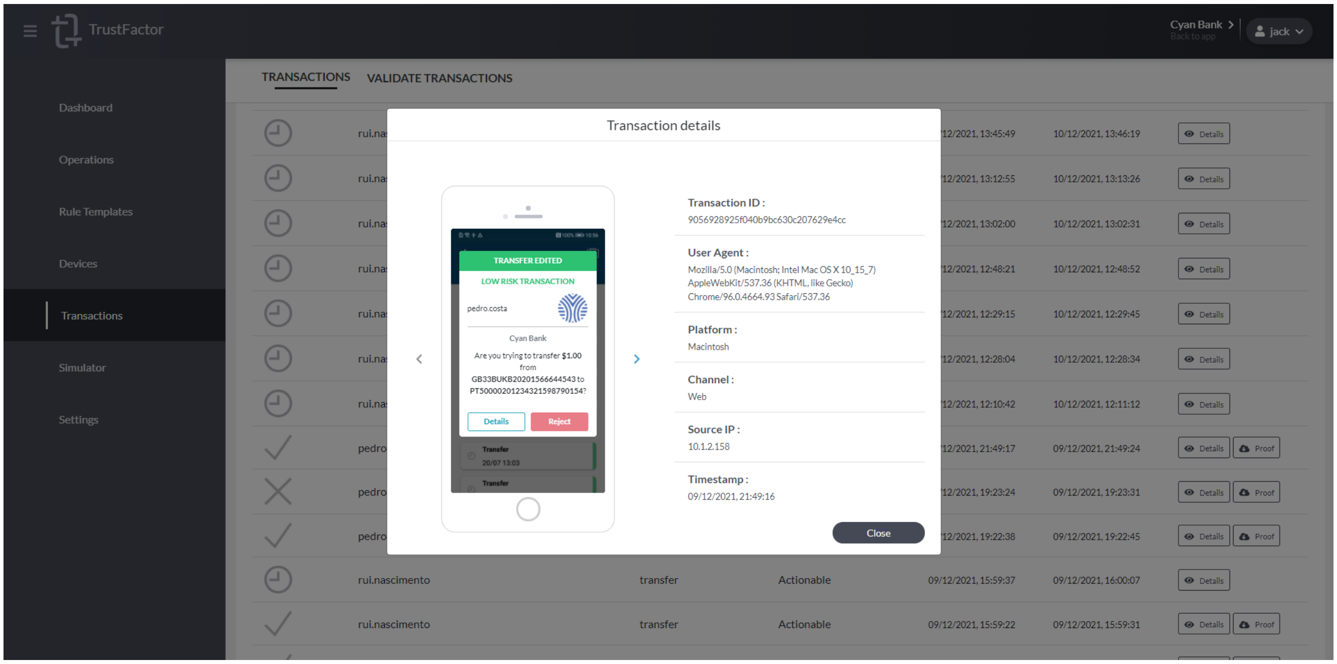


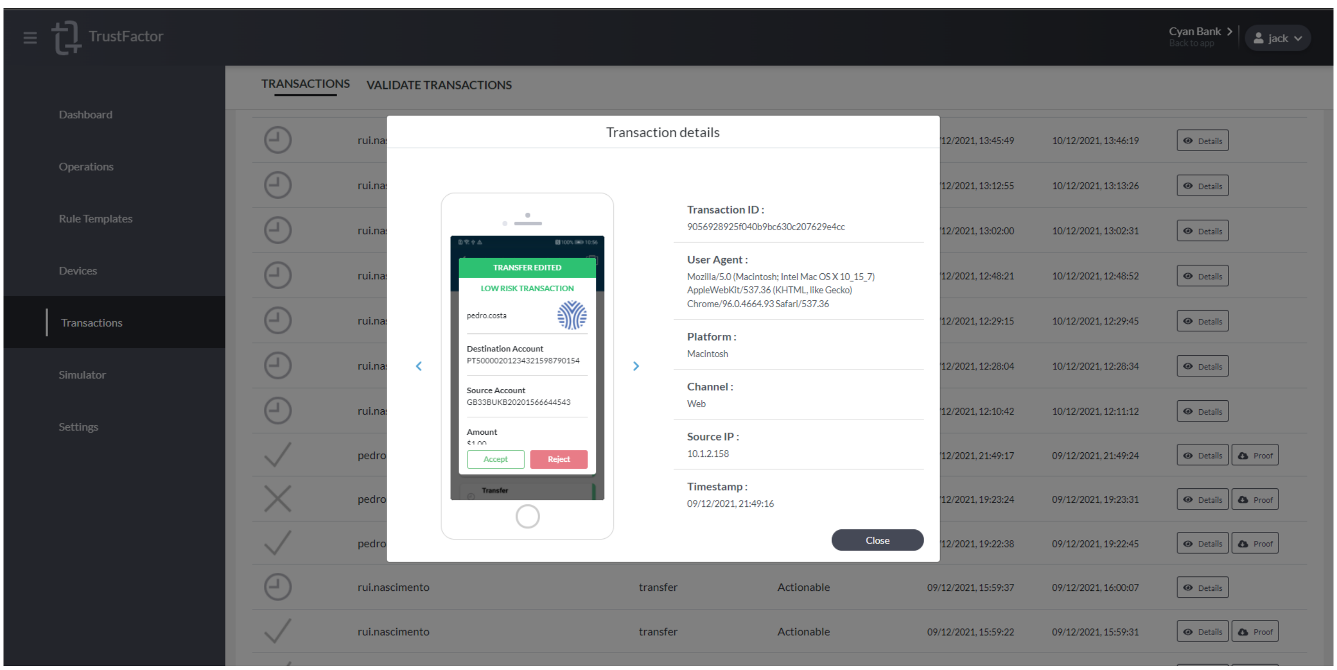
The transactions tab allows backoffice users to view the authentication history of the application users.

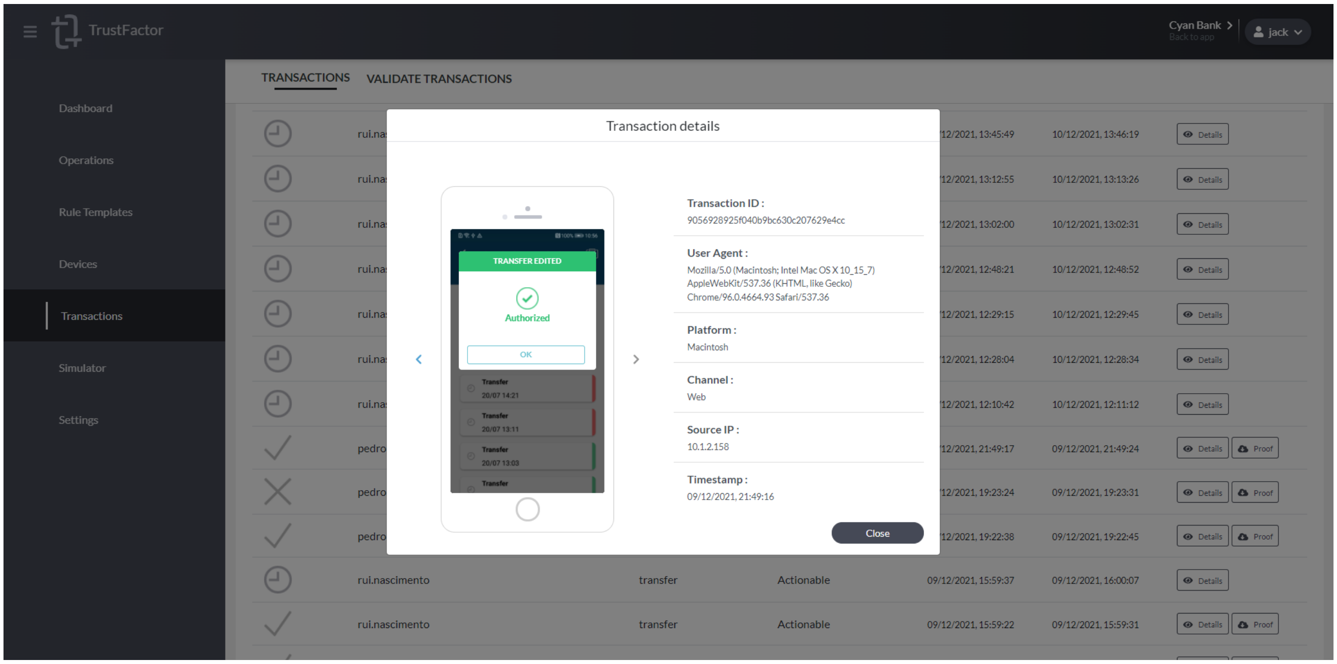
You can filter for:

- Transaction ID
- App User Unique ID
- Contract Key
- Action
- Type
- Status
- Created At (time period)

By pressing the "Details" button at the end of the transaction line in the table, you can access a mockup of what the user saw when they interacted with the authentication request.

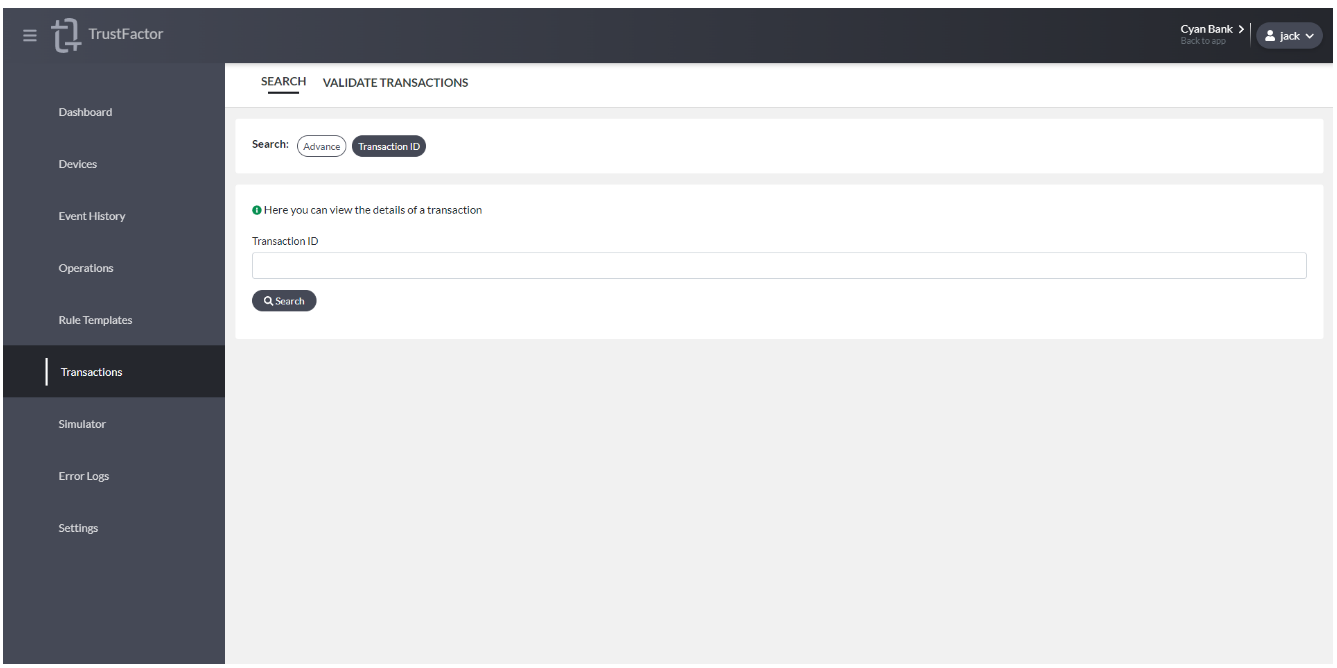

TrustFactor
Cyan Bank
Back to app
jack
Dashboard
Operations
Rule Templates
Devices
Transactions
Simulator
Settings
TRANSACTIONS
VALIDATE TRANSACTIONS
Transaction details
TRANSFER EDITED
LOW RISK TRANSACTION
pedro.costa
Cyan Bank
Are you trying to transfer $1.00 from GB33BUKB20201566644543 to PT50000201234321598790154?
Details
Reject
Transaction ID :
9056928925f040b9bc630c207629e4cc
User Agent :
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 Safari/537.36
Platform :
Macintosh
Channel :
Web
Source IP :
10.1.2.158
Timestamp :
09/12/2021, 21:49:16
Close
rui.nascimento
transfer
Actionable
09/12/2021, 15:59:37
09/12/2021, 16:00:07
09/12/2021, 15:59:22
09/12/2021, 15:59:31
Details
Proof

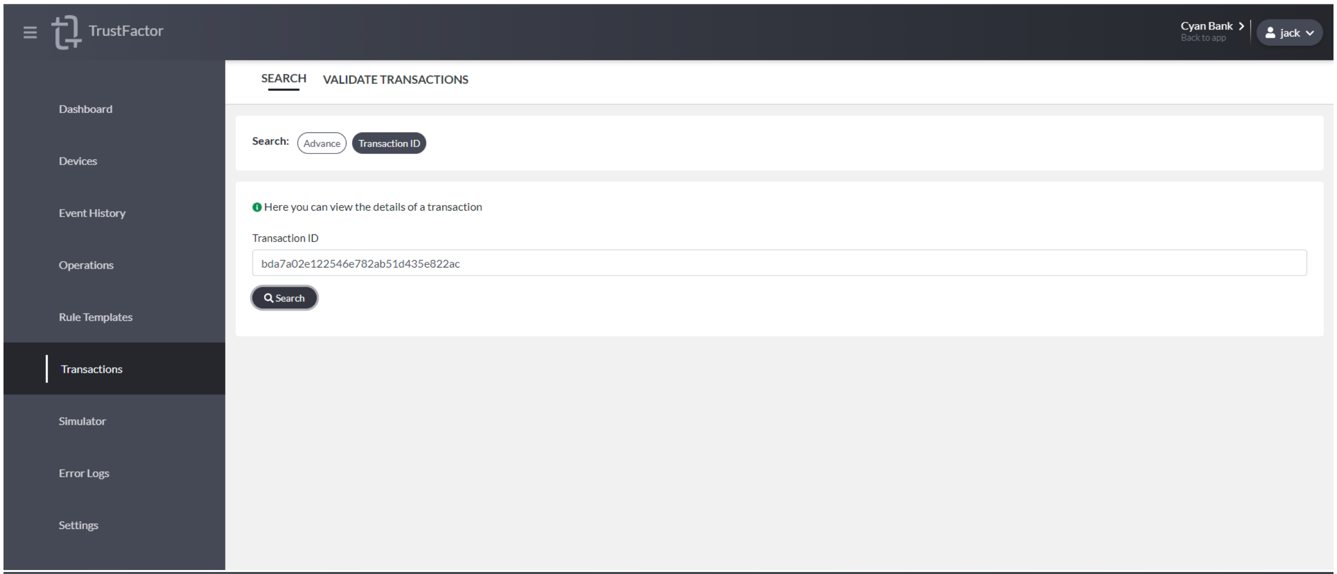

TrustFactor
Cyan Bank
Back to app
jack
Dashboard
Operations
Rule Templates
Devices
Transactions
Simulator
Settings
TRANSACTIONS
VALIDATE TRANSACTIONS
Transaction details
TRANSFER EDITED
LOW RISK TRANSACTION
pedro.costa
Destination Account
PT50000201234321598790154
Source Account
GB33BUKB20201566644543
Amount
Accept
Reject
Transaction ID :
9056928925f040b9bc630c207629e4cc
User Agent :
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.93 Safari/537.36
Platform :
Macintosh
Channel :
Web
Source IP :
10.1.2.158
Timestamp :
09/12/2021, 21:49:16
Close
rui.nascimento
transfer
Actionable
09/12/2021, 15:59:37
09/12/2021, 16:00:07
09/12/2021, 15:59:22
09/12/2021, 15:59:31
Details
Proof

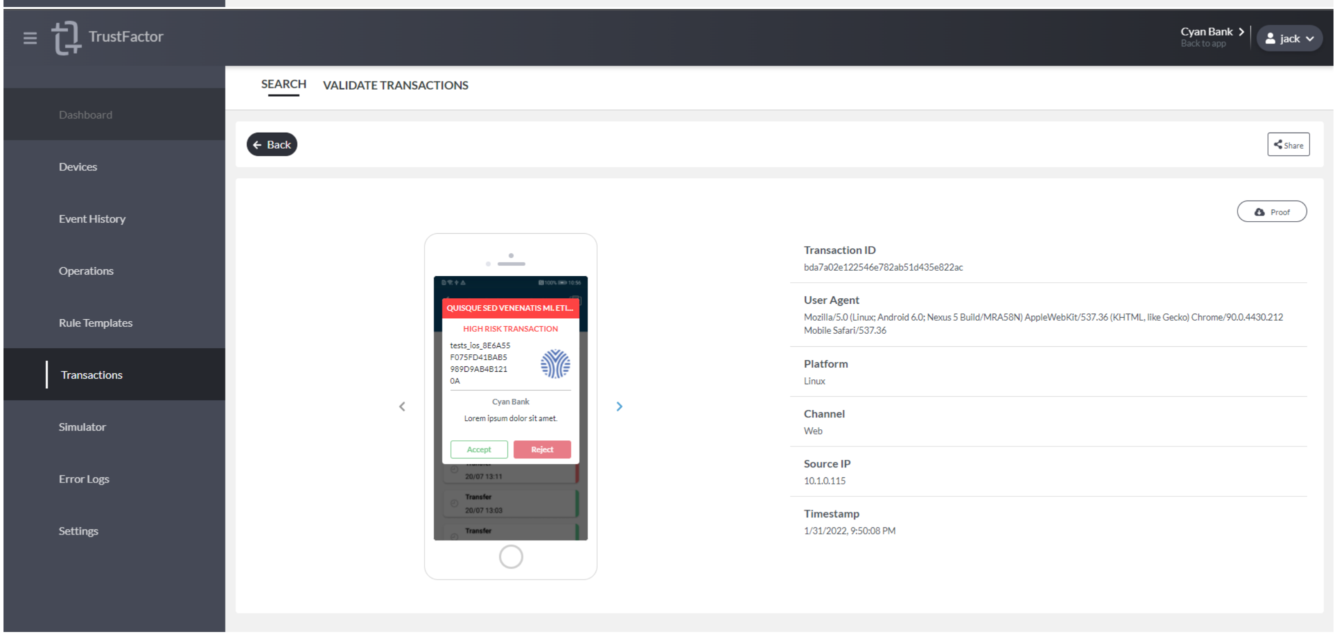


Transactions Search - By Transaction ID

Permissions required to view this screen: - **View Transactions**

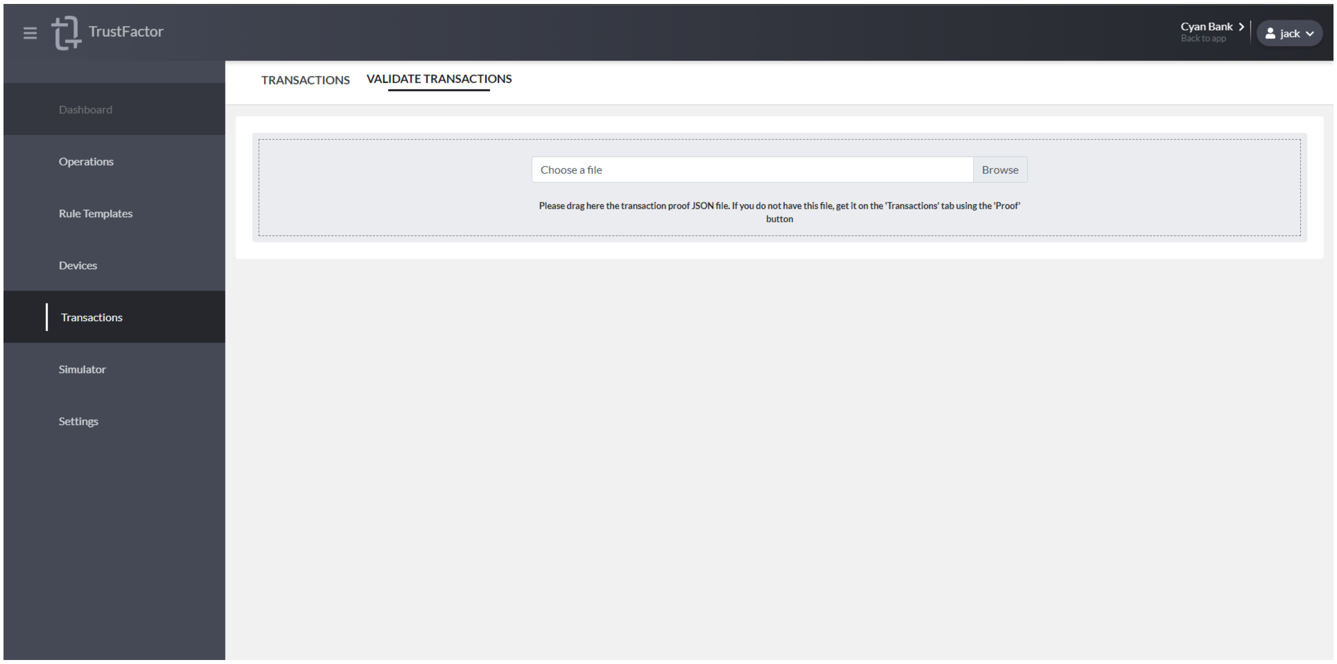




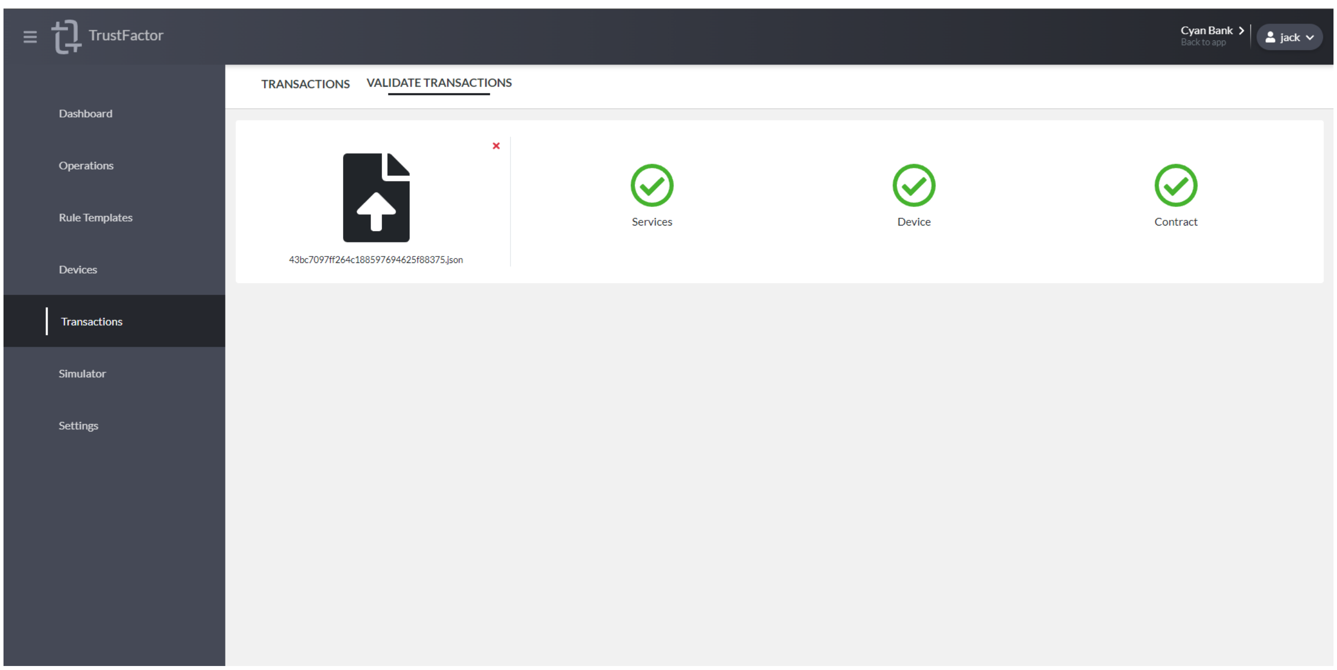


Validate Transactions

Permissions required to view this screen: - **Transaction Proofs Reader**



The "validate transactions" tab allows to validate a transaction proof.

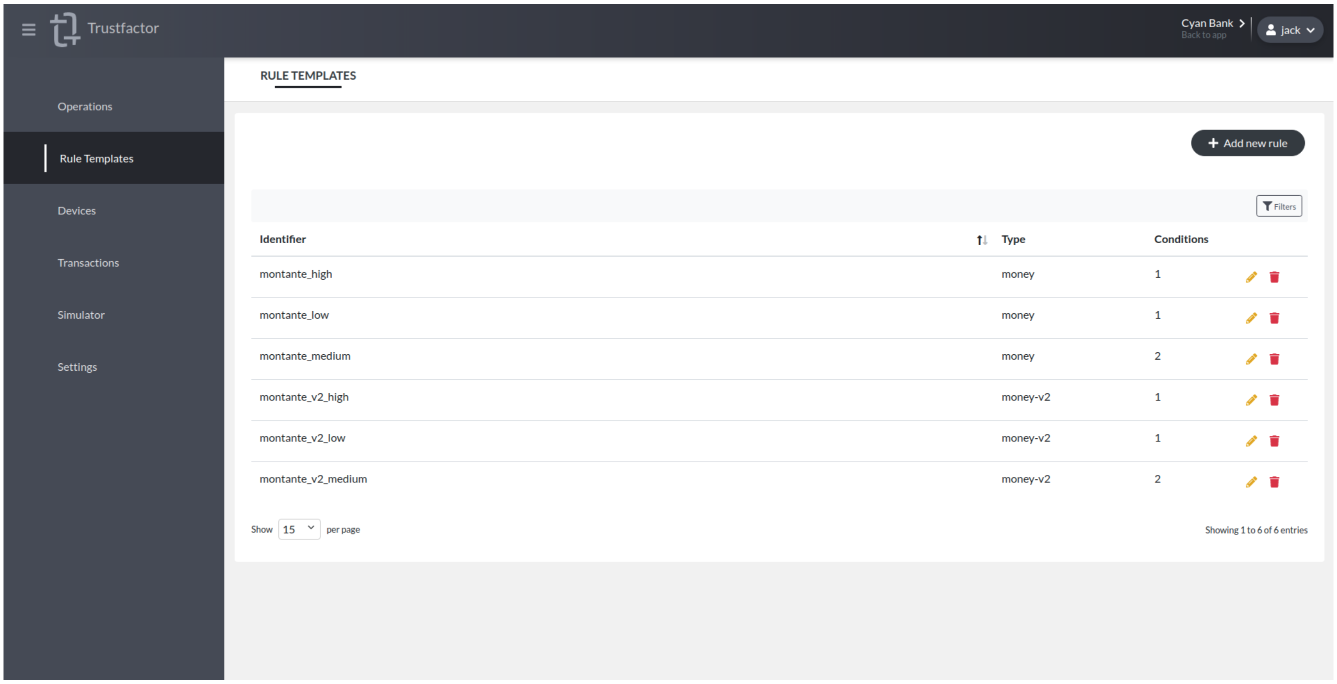


When the file is uploaded, some validations will be done to confirm that the transaction proof is valid.

Rule Templates

Permissions required to view this screen: **- View Rules**

Permissions required to make changes in this screen: **- Manage Rules**



The rule templates tab can be used to create templates for risk rules that can be applied to your operation parameters.

Risk rules must match the same type for each risk rating (low, medium and high). Different parameter types allow for different conditions.

Example - Funds Transfer

Let's explore an example:

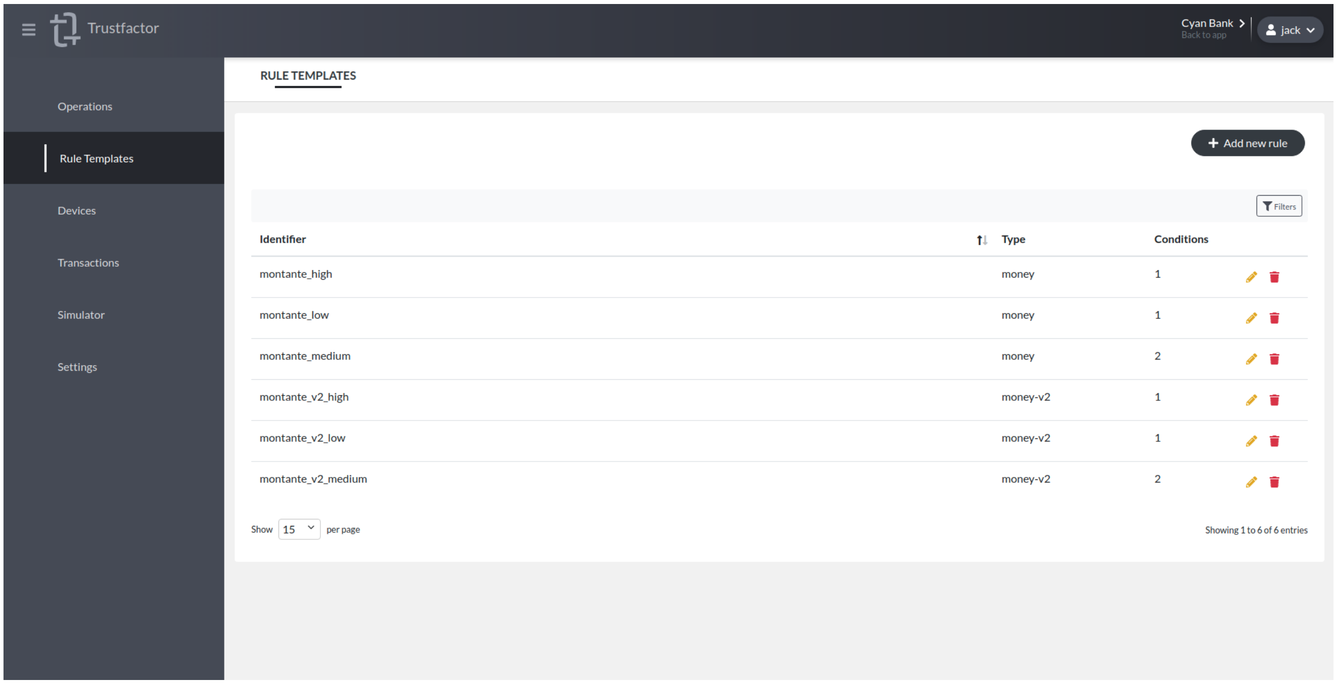
A banking application wants to define a low, medium and high risk rules for funds transfers based on the amount of the transaction. Their definition is as follows:

- API call to transfer funds has 3 parameters:
  - Source Account
  - Destination Account
  - Transaction Amount
- Transaction Amounts below 500 USD or 500 EUR are ranked **low risk**
- Transaction Amounts between 501 USD and 1500 USD or 501 EUR and 1000 EUR are ranked **medium risk**
- Transaction Amounts above 1501 USD or 1001 EUR are ranked **high risk**
- Transaction Amounts in **other currencies should not be subject to risk calculation**

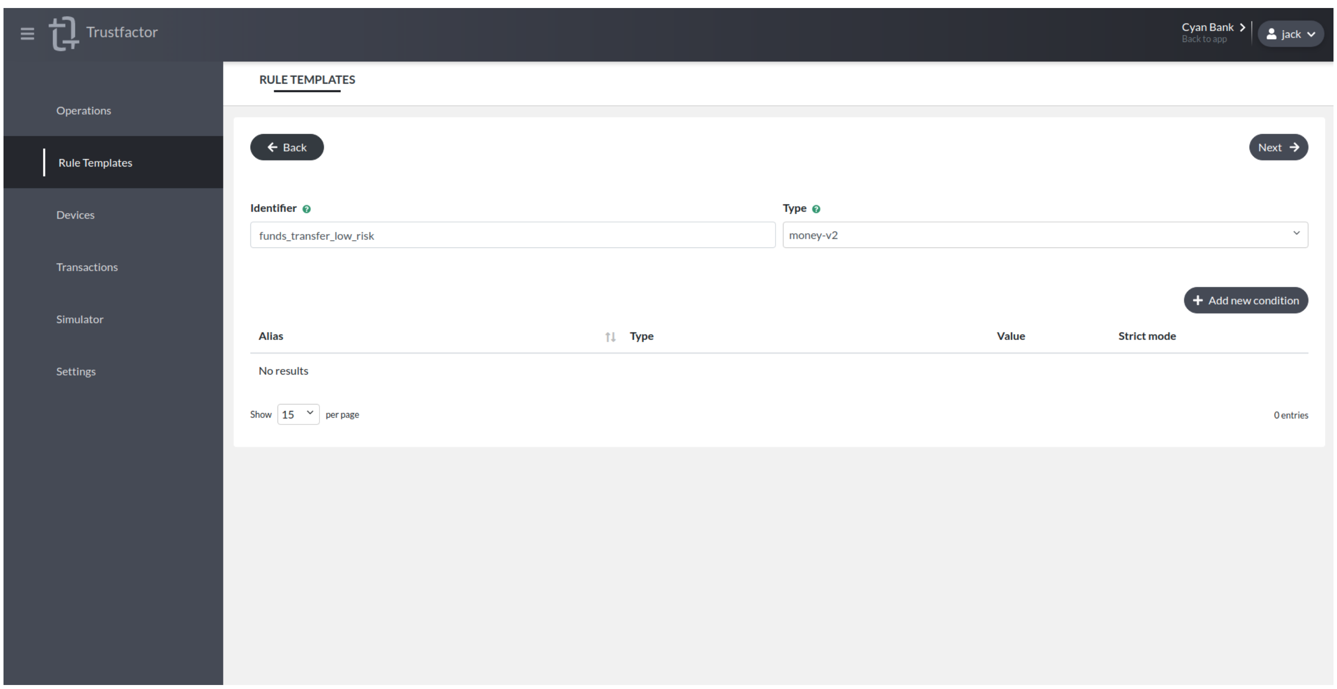
Because we are using money (amount and currency) for our definition, we will start by using *Money Rules*.

Money Rules

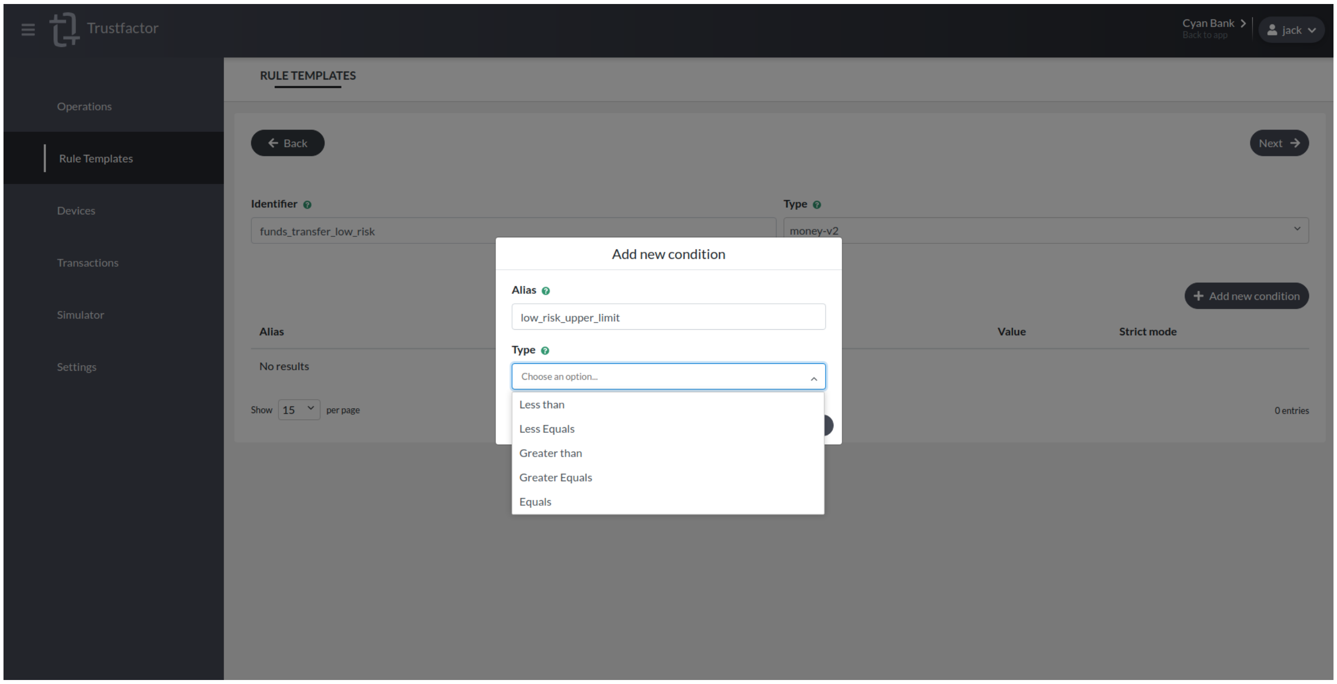
In order to create these rules and then use them in operations, we must first enter the Rule Templates tab, and press *+ Add new rule*.



In the next screen, we enter the risk rule identifier (funds_transfer_low_risk) and define the type of the parameter this rule will apply to (money-v2).



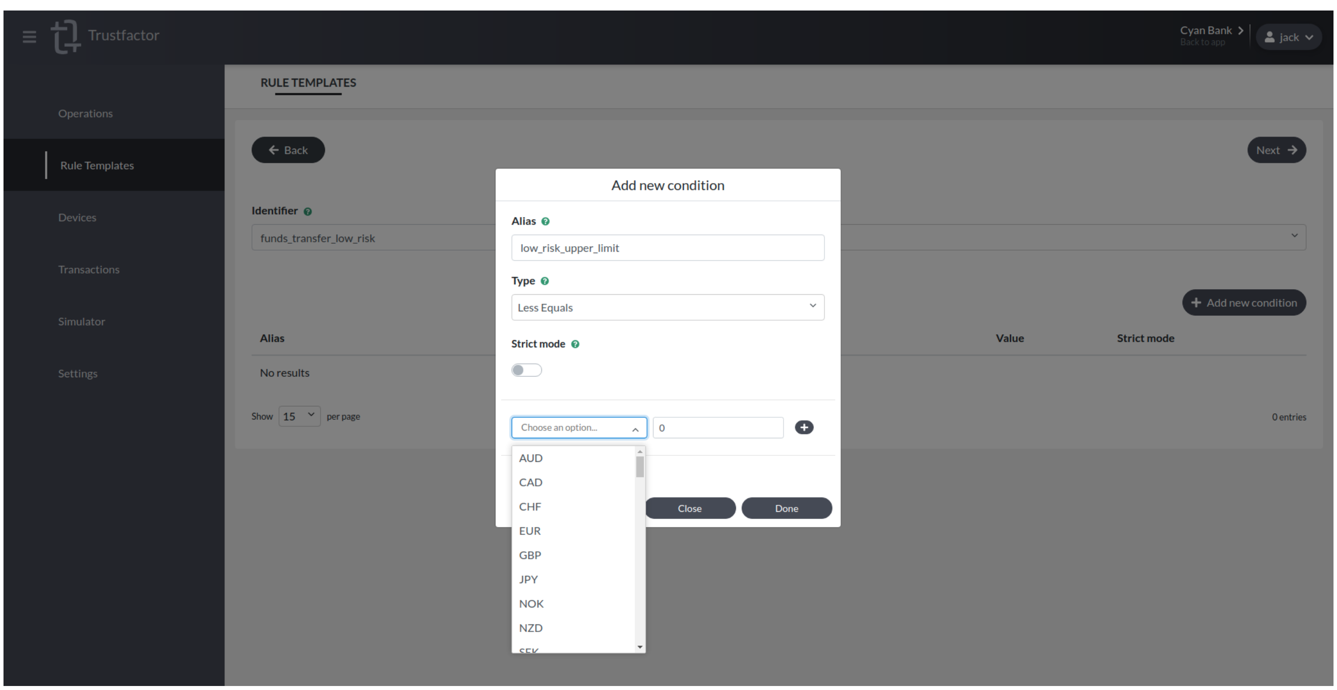
After these two configurations are done, we can start adding conditions to our rule by pressing *+ Add new condition*.



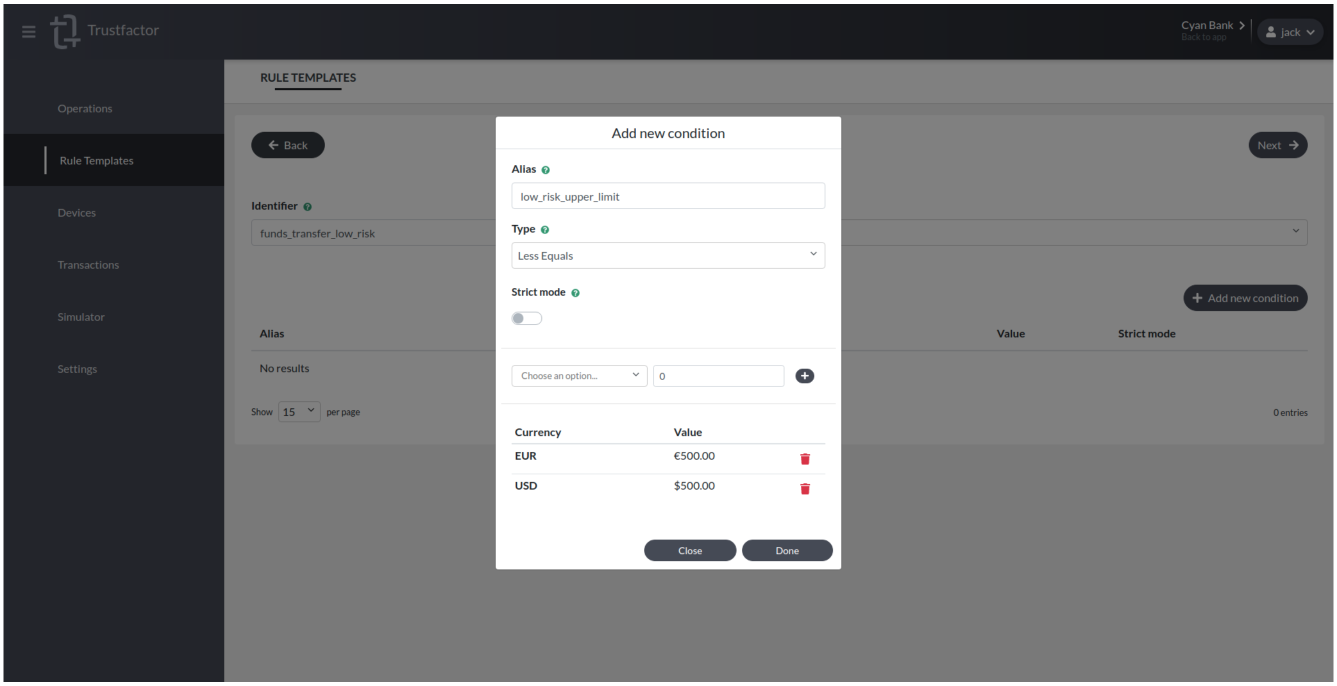
In the pop-up, we define the name of alias for the rule (low_risk_upper_limit) and then set it as "Less Equals". This will change the pop-up UI to allow you to enter different amount / currency pairs. The available logic operators are:

- **Less equals** (equivalent to <=)
- **Less than** (equivalent to <)
- **Greater than** (equivalent to >)
- **Greater equals** (equivalent to >=)
- **Equals** (equivalent to =)

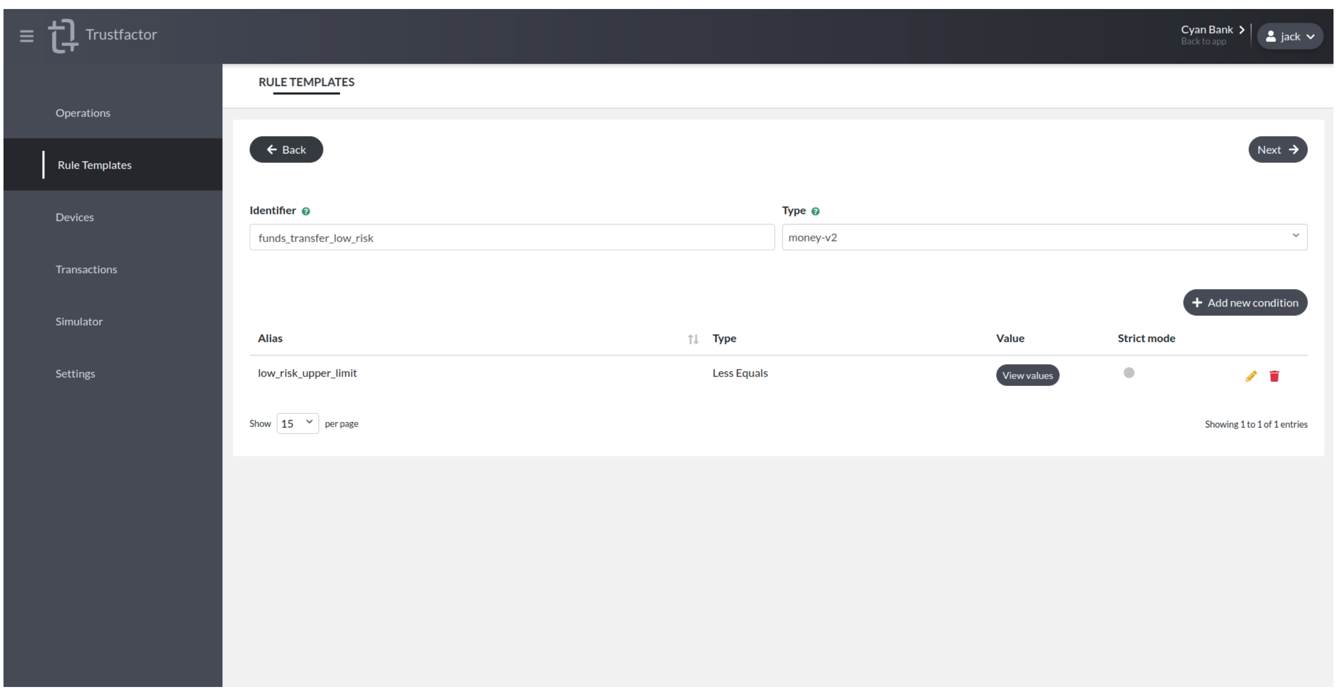
You may also notice the Strict mode is disabled – this ensures the rule is compliant with the last statement in the definitions above. In Strict Mode, if a new authentication request is created in a currency that has not been defined in the risk rules, it will throw an error. If Strict Mode is disabled, risk rules are not applied when the authentication request uses a transaction that has not been defined in the Rule Template.



After you are done entering your limits for each currency, you can press *Done*.



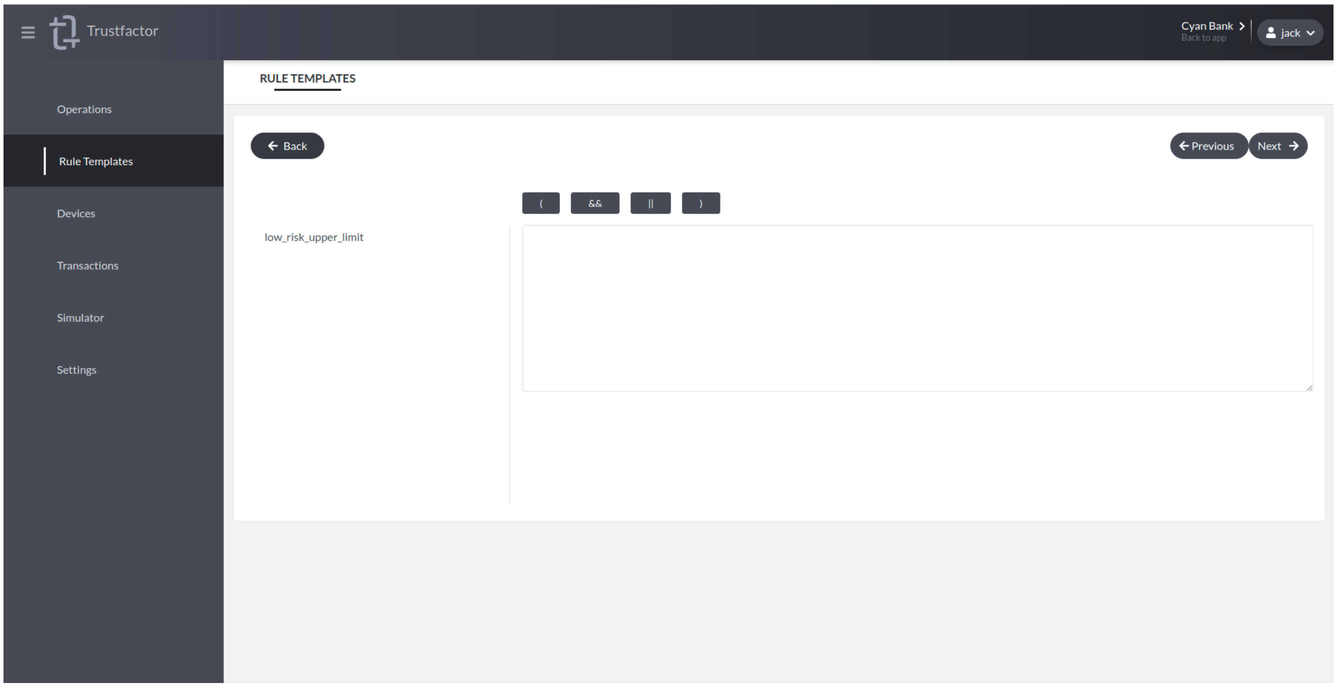
This will add the condition you just defined to the condition list. For the low risk rule, this is the only condition we need because it already implements the first statement in the definitions, so we can now press *Next* to proceed to the rule logic screen.

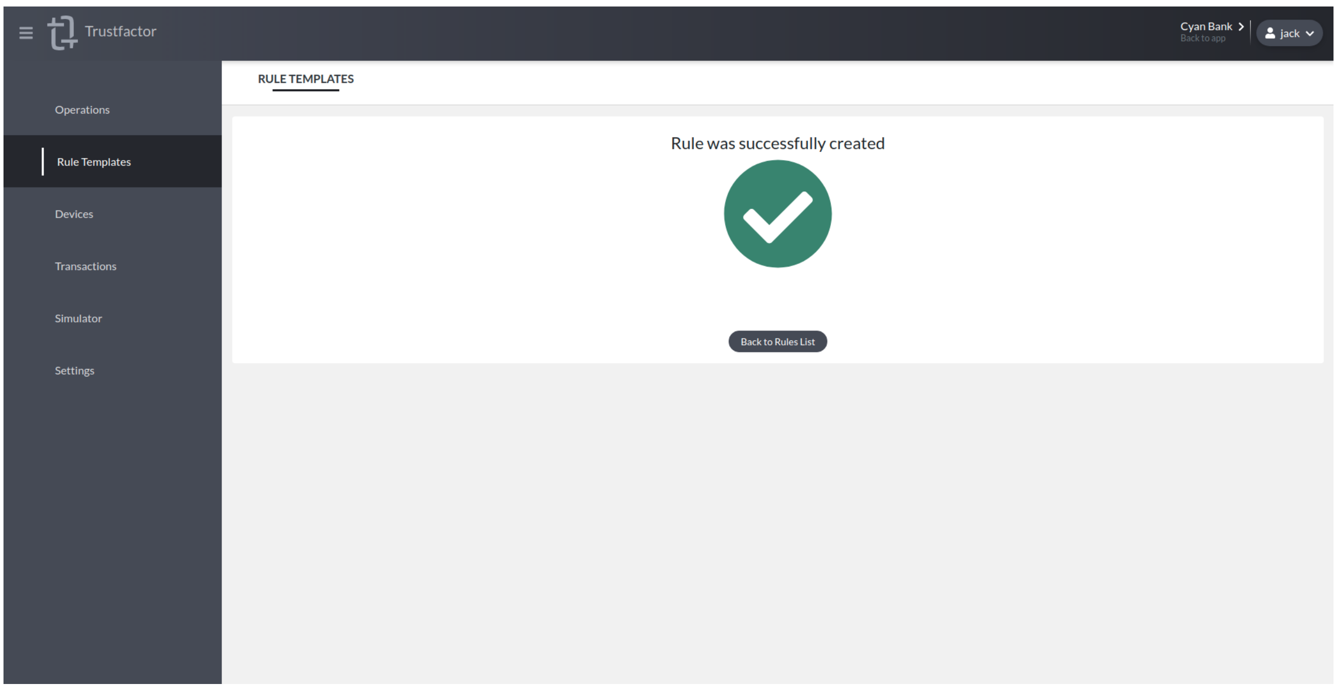


In the Rule Logic screen, we have to select the logic with which to apply our conditions. When using multiple conditions, we can define their relationship through logic operators. The supported operators are:

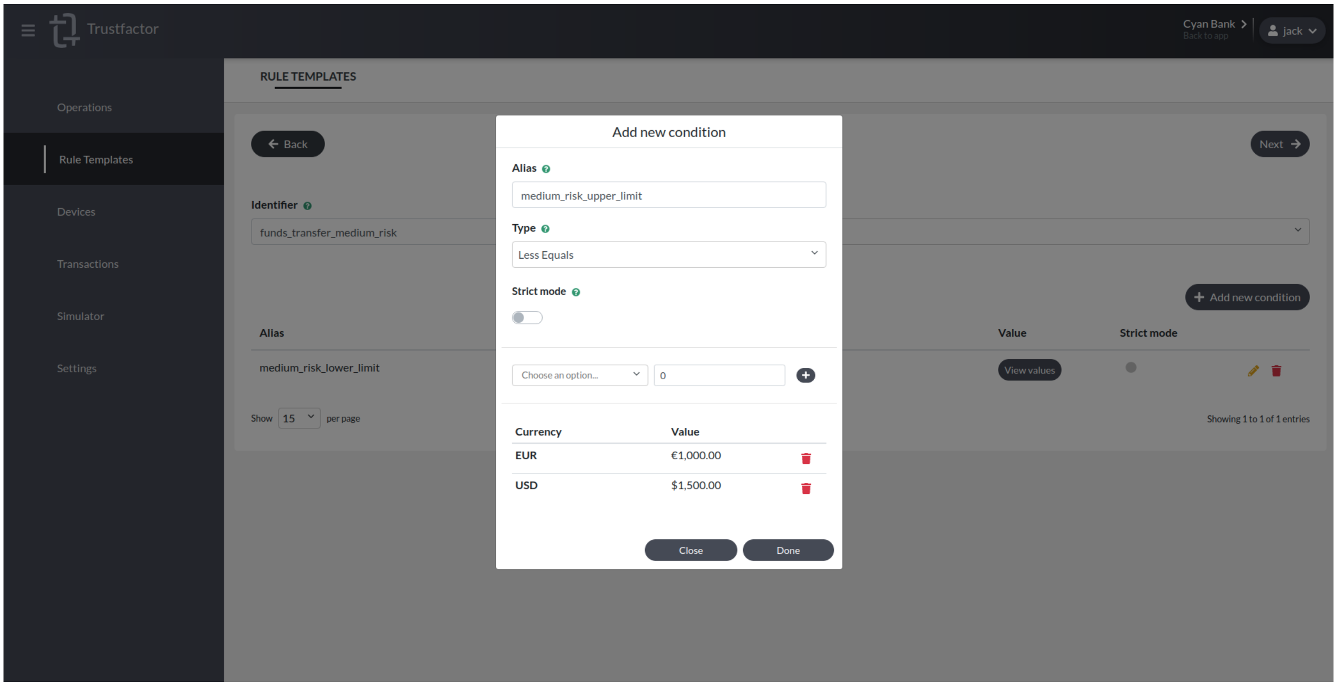
- **AND** (&&)
- **OR** (||)

In this case it's simple because we only have one condition, so we just select that *low_risk_upper_limit* condition from the left by clicking on it and moving it to the rule logic pane. After that, press *Next* again and if all goes well with the validation, we should see a success screen and our first rule has been created.

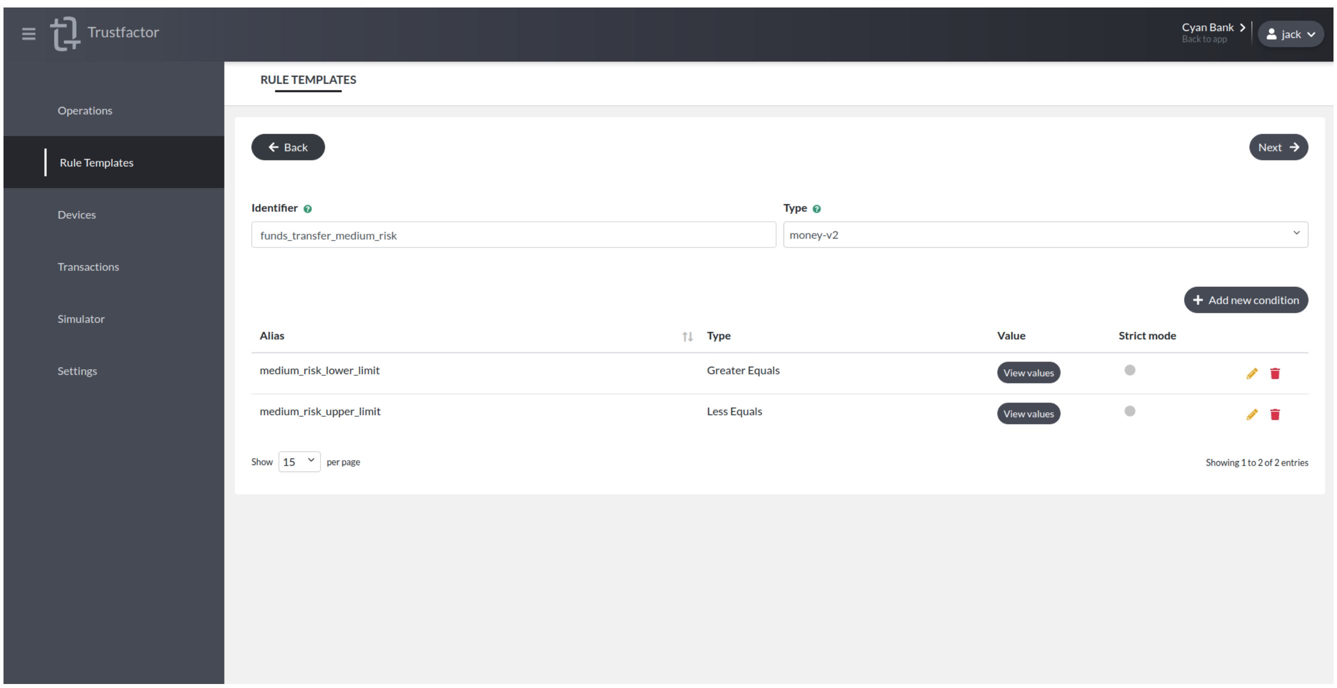




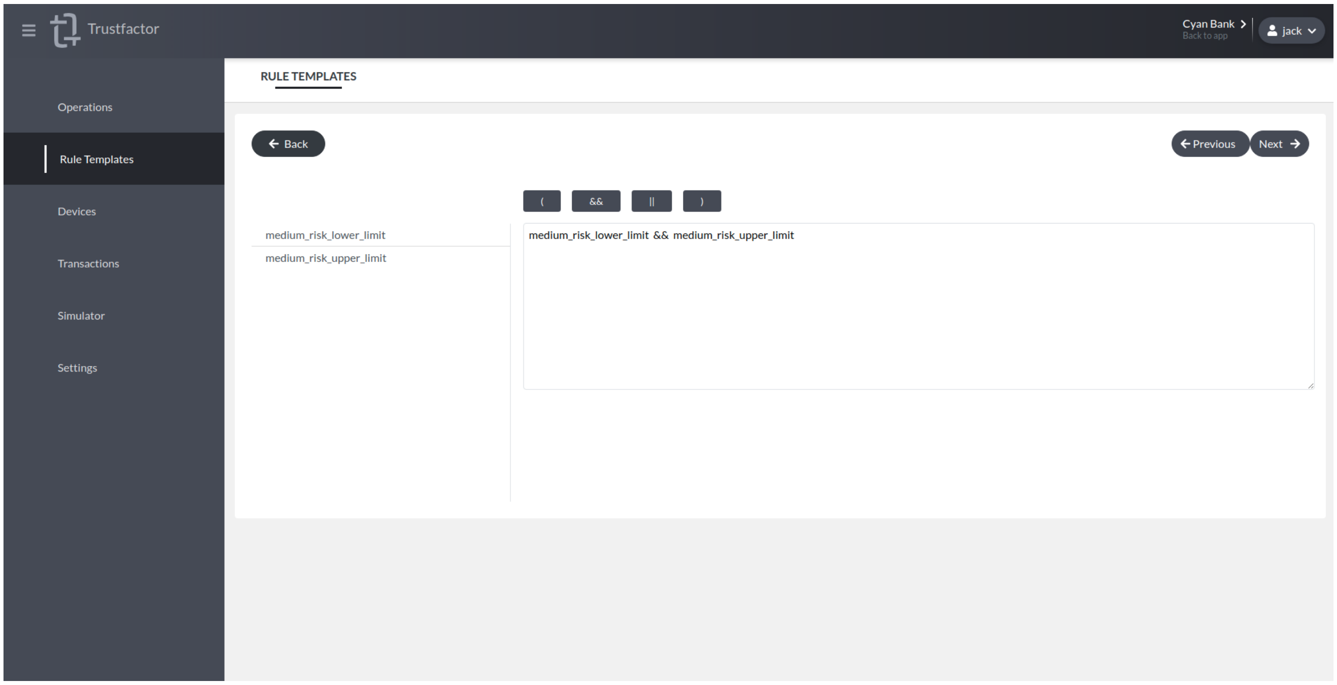
Now that we have a rule for the first definition (low risk), we need to create the second one for medium risk. We follow the same steps as before but now we need to add two separate conditions, one "greater equals" and another "lower equals" to define the upper and lower bounds of the values that match this rule.



After both conditions are entered you should see a screen like the one below.



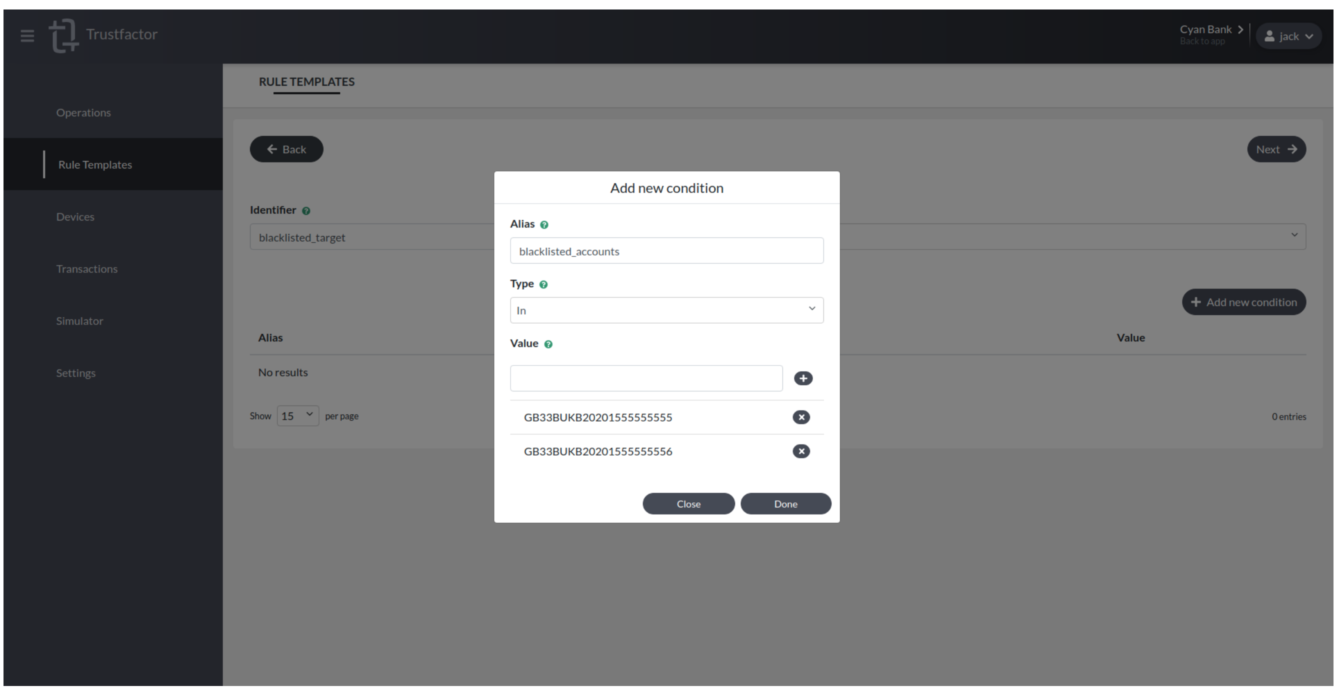
Now in the rule logic screen, we need to reference both rules and say we want both of them to match through an **AND** operator.



After this, we press *Next* to proceed to the validation screen and then *Submit* our changes to create the Medium risk Rule.

The creation of the last rule to match the definitions above is similar to the first one except instead of having an upper boundary it has a lower boundary and all values above that boundary must be rated **High Risk**.

String Rules



For string parameters, we have the following types of conditions:

- **In**

Checks whether the value matches a provided list of values

- **Contains**

Checks whether the value contains a provided value as substring

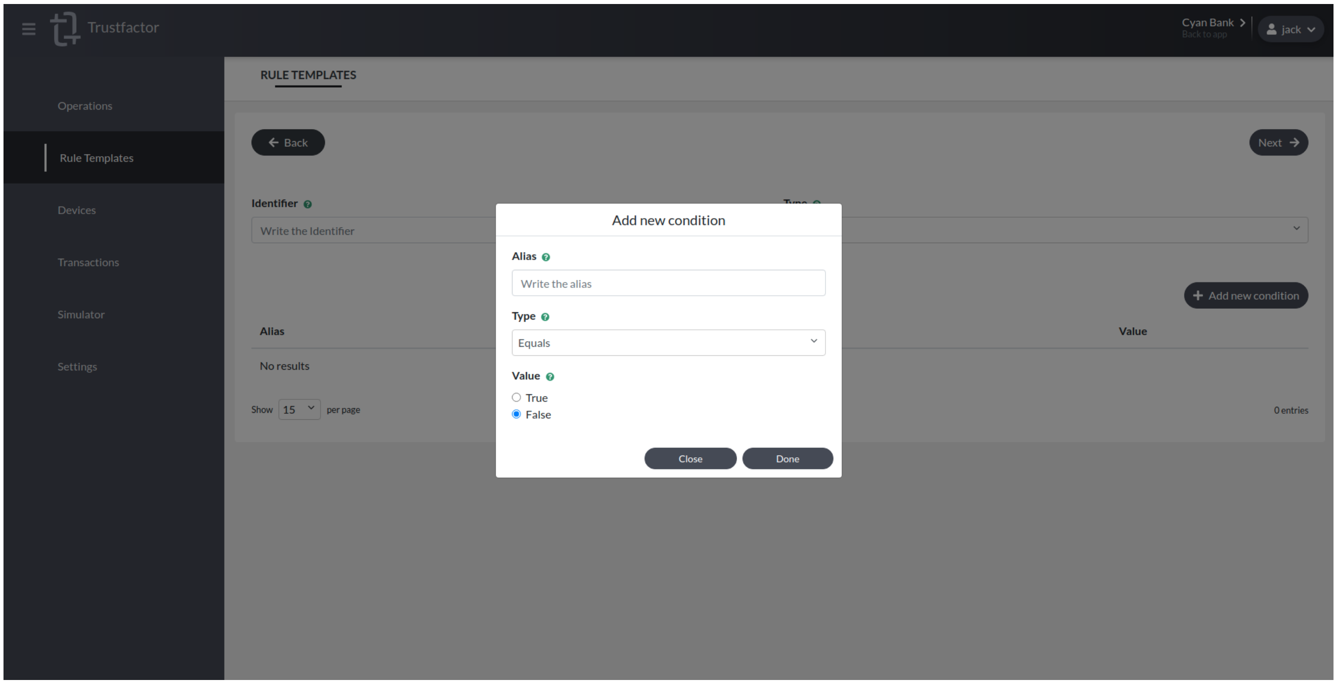
- **Equals**

Checks if the value is equal to a provided value

- **Regex**

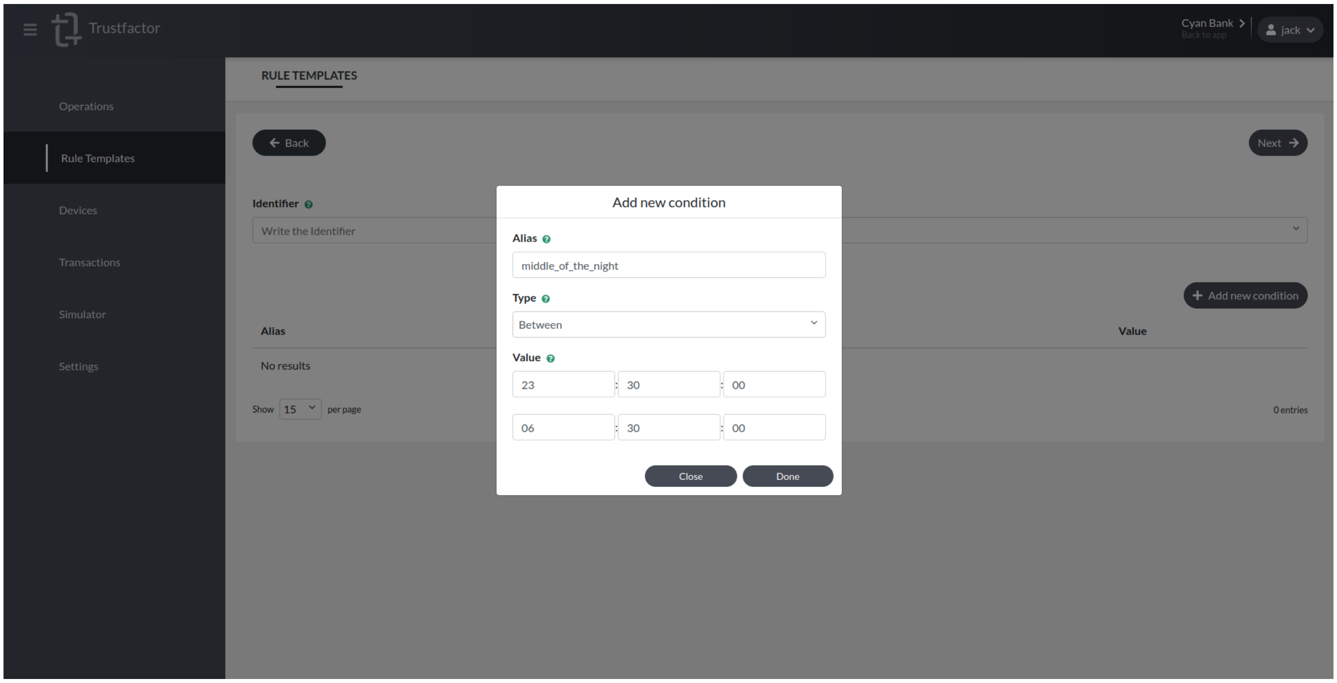
Performs a regular expression check against the value

Boolean Rules



You can also create rules that match boolean parameters. In this case, only "Equals" is supported and values can be "True" or "False".

Datetime Rules



Datetime parameters are also supported in TrustFactor and as such you can create risk rules for them. In the image above we define an interval between 11:30pm and 06:30am called "middle of the night". We can use this to raise the risk level of authentication requests performed during this time period.

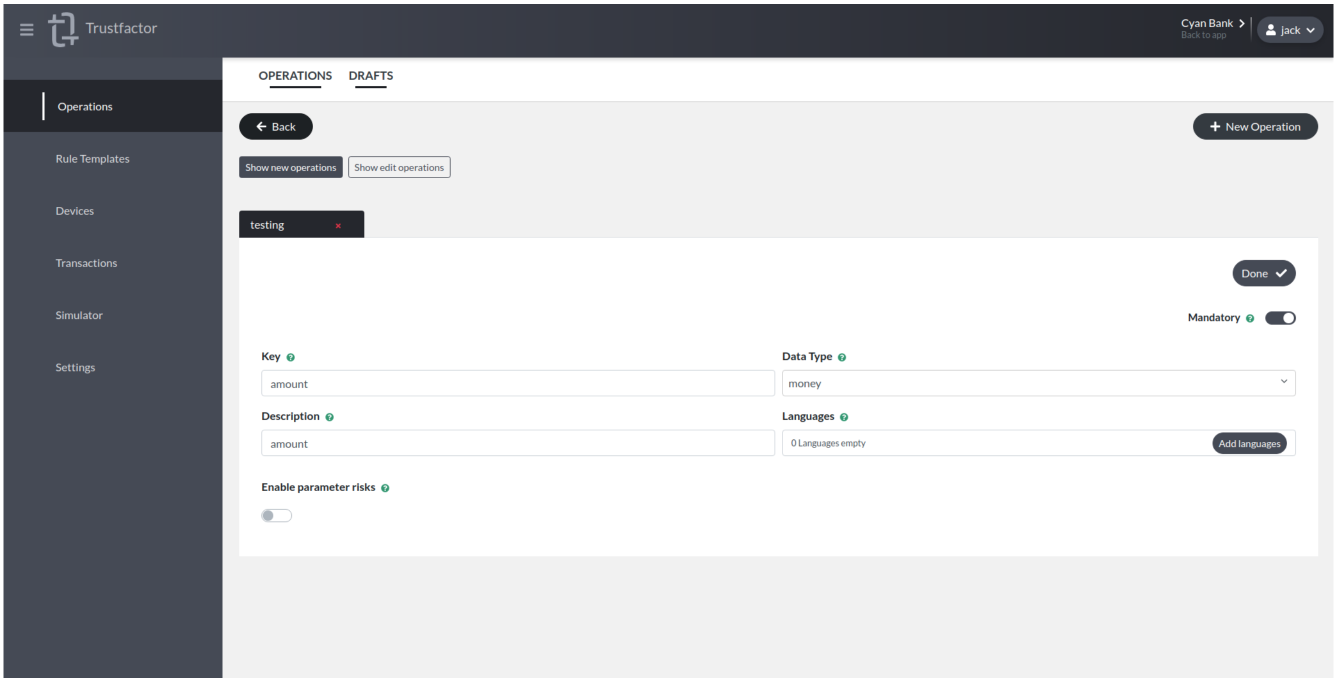
Float Rules

You can apply the same kind of rules and conditions described in the **Money Rules** above to parameters of *float* type.

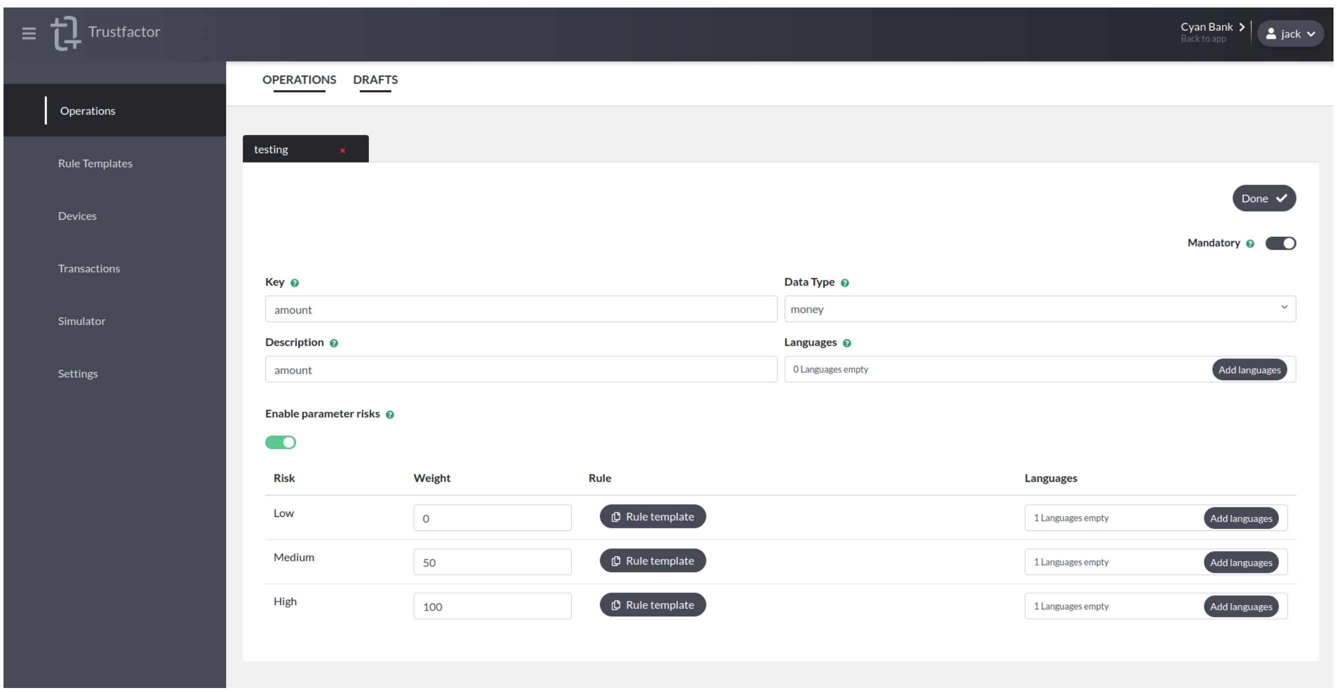
Assigning Rules to Operations

In order to assign a rule template to an operation, first you need to have an operation with a parameter of the same type as the rule you want to assign.

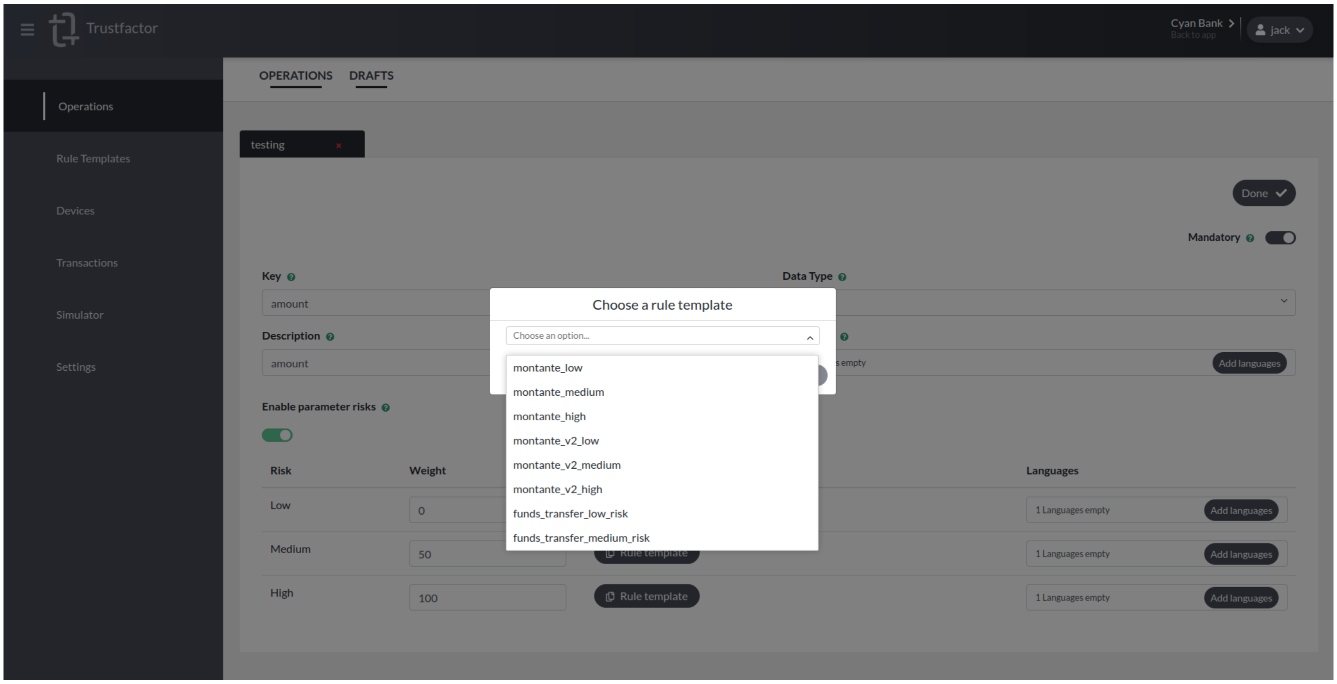
An operation parameter without risk rules looks like this:



You need to *Enable parameter risks* in order to define the risk rules for this *amount* parameter.

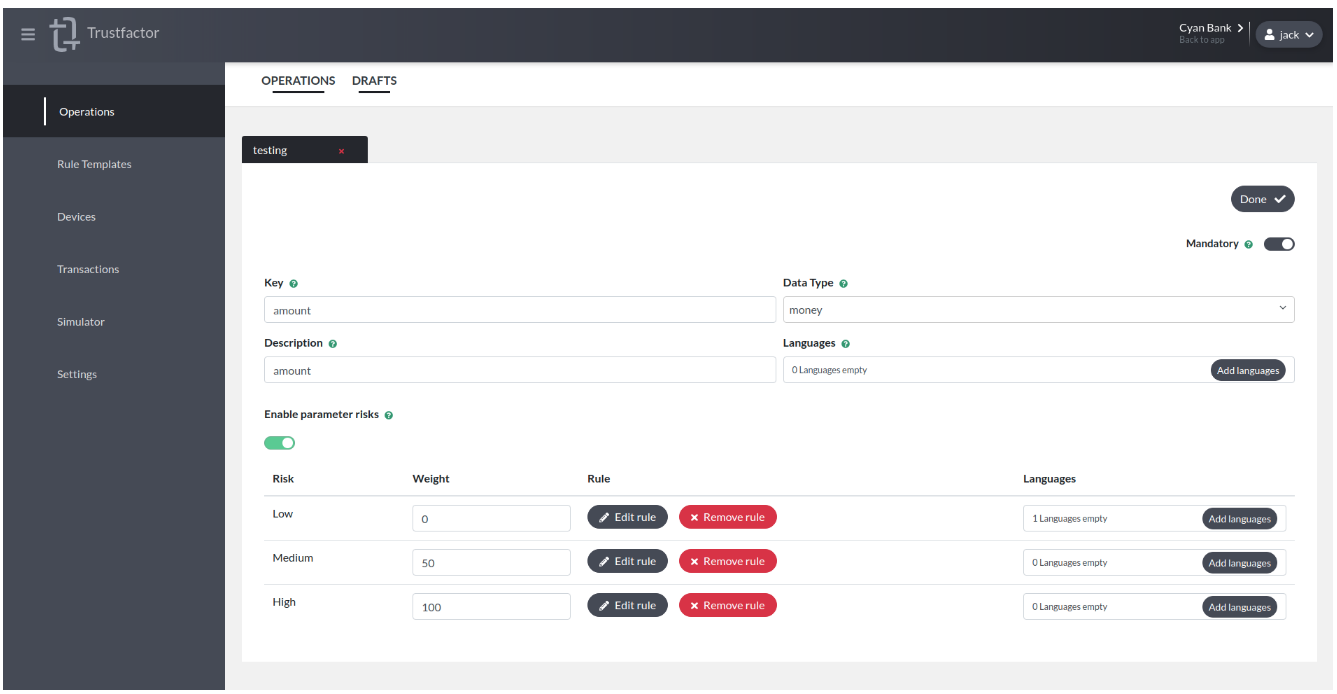


Now we can select the rules we created earlier for our example scenario from the dropdown menu at each risk level:



In addition, we can define the risk message shown to the end user when this rule matches by clicking "Add languages".

When all risk levels have an associated risk rule, you can click *Done* to complete the definition for this operation parameter. The screen should look like the one below:

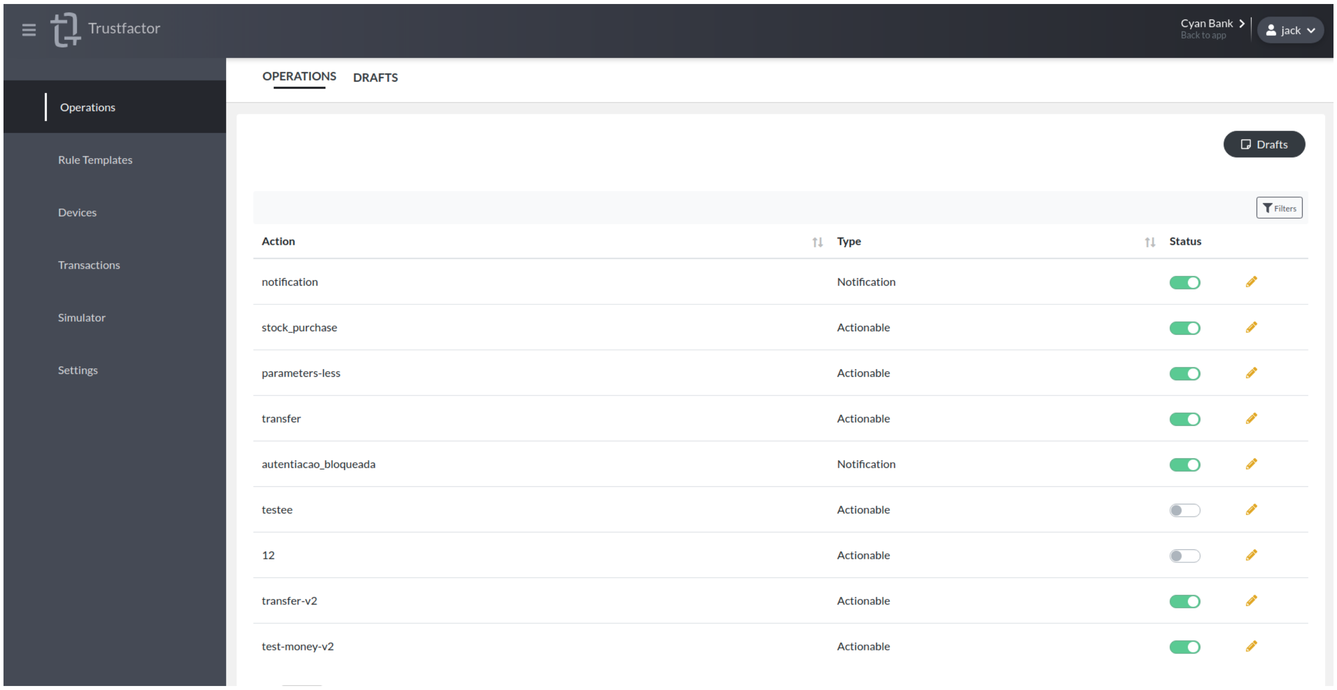


For more information on how to define Operations, read the section below.

Operations

Permissions required to view this screen: - **View Operations**

Permissions required to make changes in this screen: - **Manage Operations**



The operations menu is where you can define and manage *TrustFactor operations*. These operations will then be referenced by your application when using the *CreateTransaction* methods in the TrustFactor SDK, so you will want to create operations for sensitive actions inside your application.

Creating an operation is a relatively complex multi-step procedure, so we provide a *Drafts* screen where you can edit or create new operations before submitting them.

Drafts

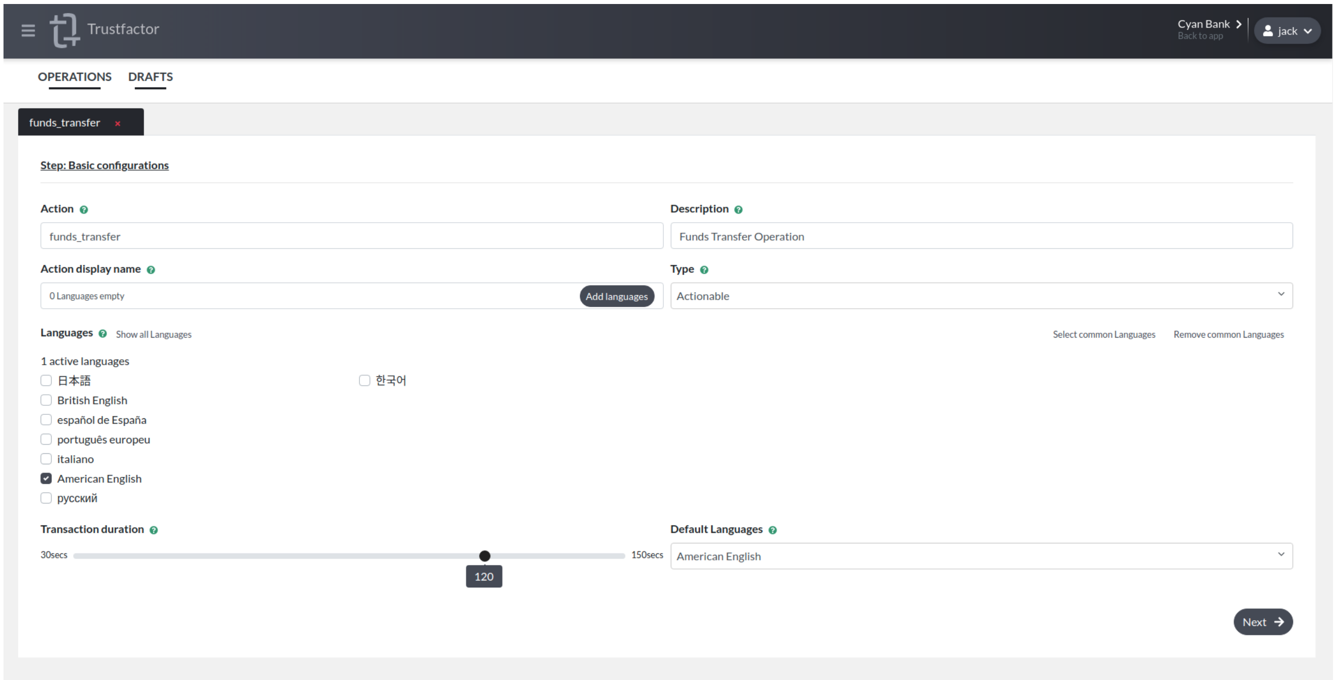
Drafts are saved automatically and allow you to resume your work later if needed. Saved Drafts are accessible by anyone with access to the application's Operations. This allows multiple users to edit the same draft, e.g. one user from the security team configures the parameters and risk rules, another user from the marketing team defines the messages shown to the user in the same operation draft.

There are two types of drafts: **New Operations** and **Edit Operations**. The toggles allow you to switch between these two types when using the Drafts.

Drafts appear under **New Operations** when you press the *+ New Operation* button to create a new operation. Drafts appear under **Edit Operations** when you press the *pencil* button to edit an existing operation.

Creating a new operation

Step: Basic Settings In this screen we define the basic settings for the operation. The first thing to notice here is the internationalization support through the Languages field. If you define multiple languages, TrustFactor will check if the end-user's device is in a language you defined and show the messages accordingly. If the device is set to an undefined language, TrustFactor will fall-back to the default language for the operation.



- **Action**

The action field is the identifier for this operation that will be used in the SDK's CreateTransaction methods. Every operation must have a unique Action within the application. This value is not shown to the end-user.

- **Action display name**

This field is the name of the operation as it is shown to the user. We need to press *Add languages* to set the name of the operation in each defined language.

- **Description**

The description field is used for notes or to describe the operation. This value is not shown to the user and is only readable through the backoffice.

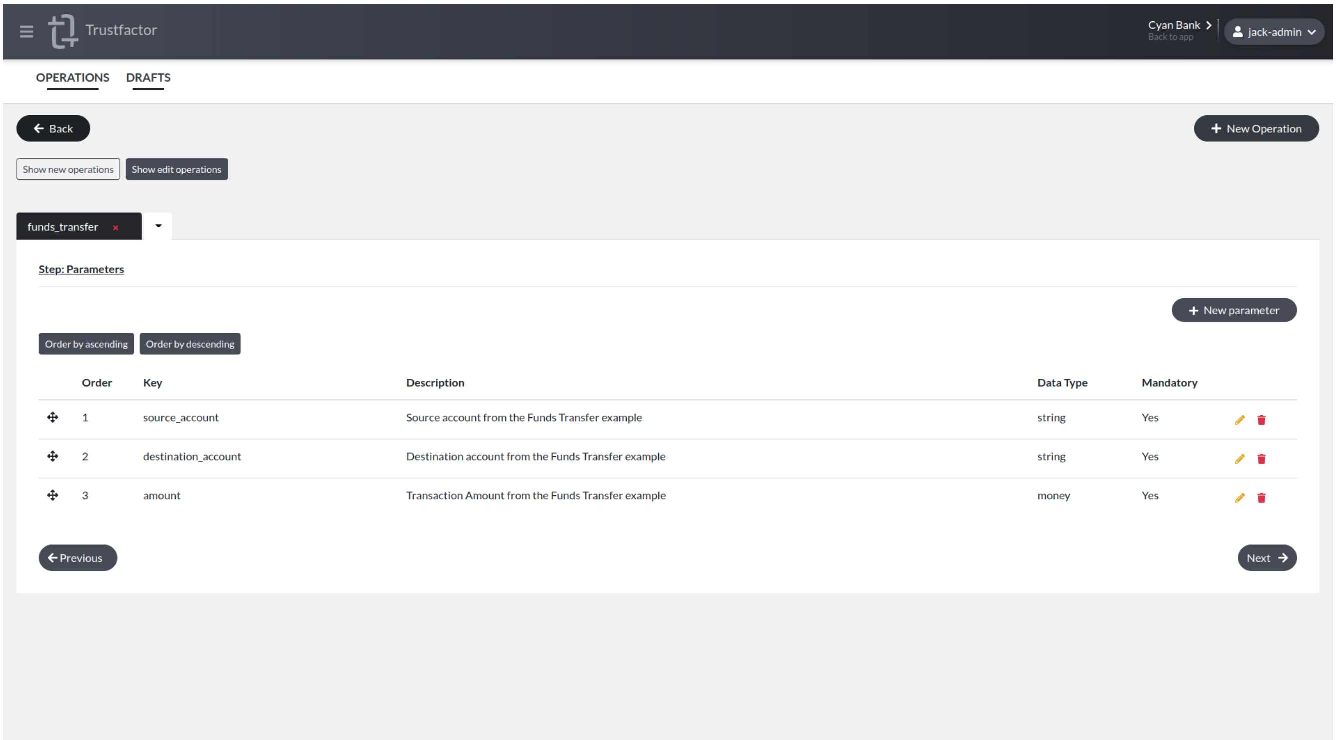
- **Type**

There are two main operation types: **Notification** and **Actionable** . **Actionable** is the type of transaction that end-users can approve or reject. This requires the callbacks to be configured properly in order to receive the authentication request result from the end-user's mobile app. **Notification** is a fire-and-forget type of authentication request. This type is useful to securely send messages to the end-user's mobile app that only have an "OK" button instead of the "Approve" and "Reject" buttons.

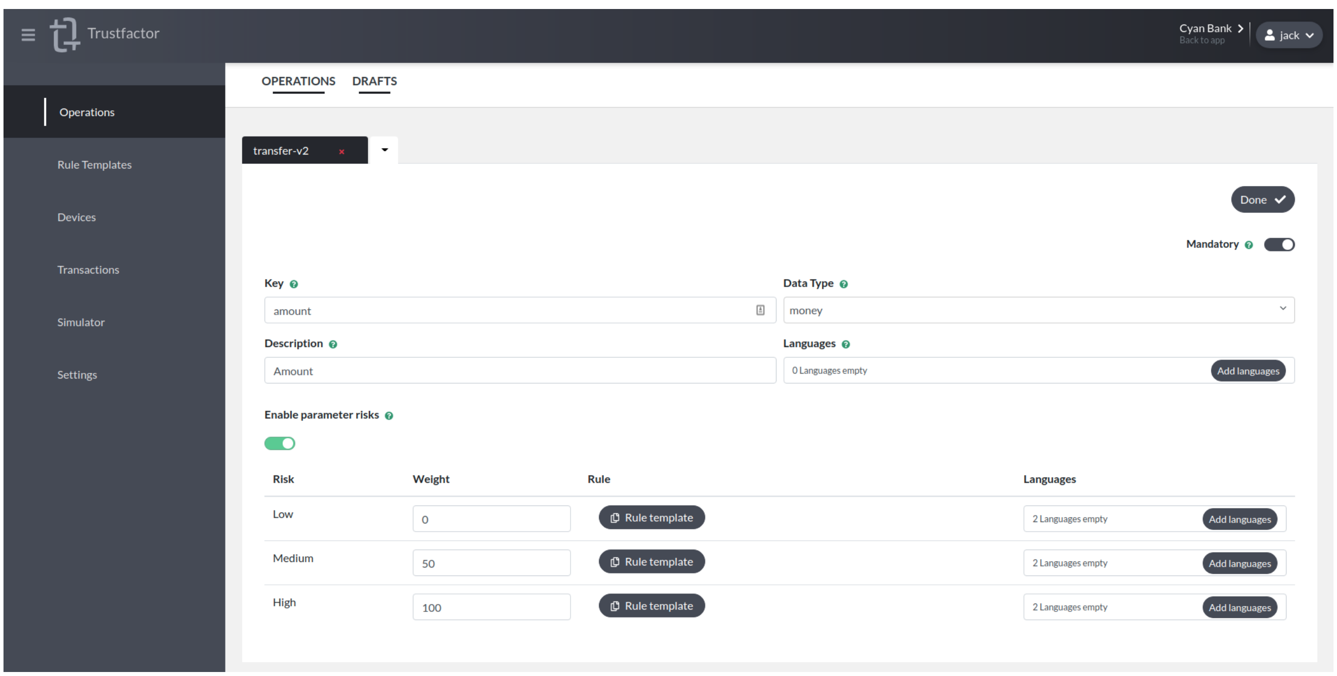
- **Default Language**

This setting defines the language TrustFactor will default to if the end-user's device is configured in a language that is not set in the **Languages** field.

Step: Parameters The **Parameters** screen is where we define the parameters for the operation and list existing parameters for the operation.



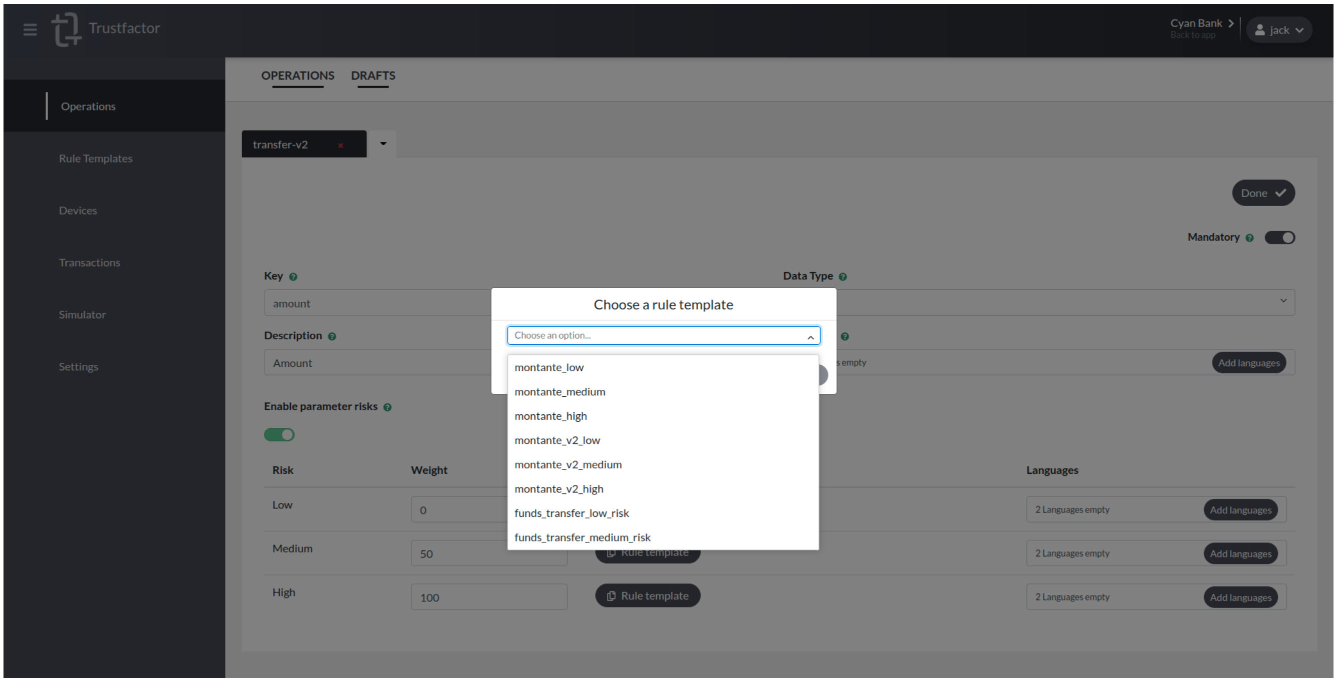
We can create new parameters by pressing + *New parameter*.



In the new parameter screen you need to define your parameter:

- **Key** defines the name of the parameter to be used when calling CreateTransaction on the SDK. This is not a translated field and will not be shown to the end-user.
- **Data Type** defines the data type for the parameter. Parameters can of four different types: *boolean*, *float*, *money* or *string*
- **Description** defines a developer-only description much like the operation's description. This is not shown to the end-user.
- **Languages** is used to define the visible translations of the parameter's name. You have to define the visible name for the parameter in every language you selected in the operation's basic setttings screen.

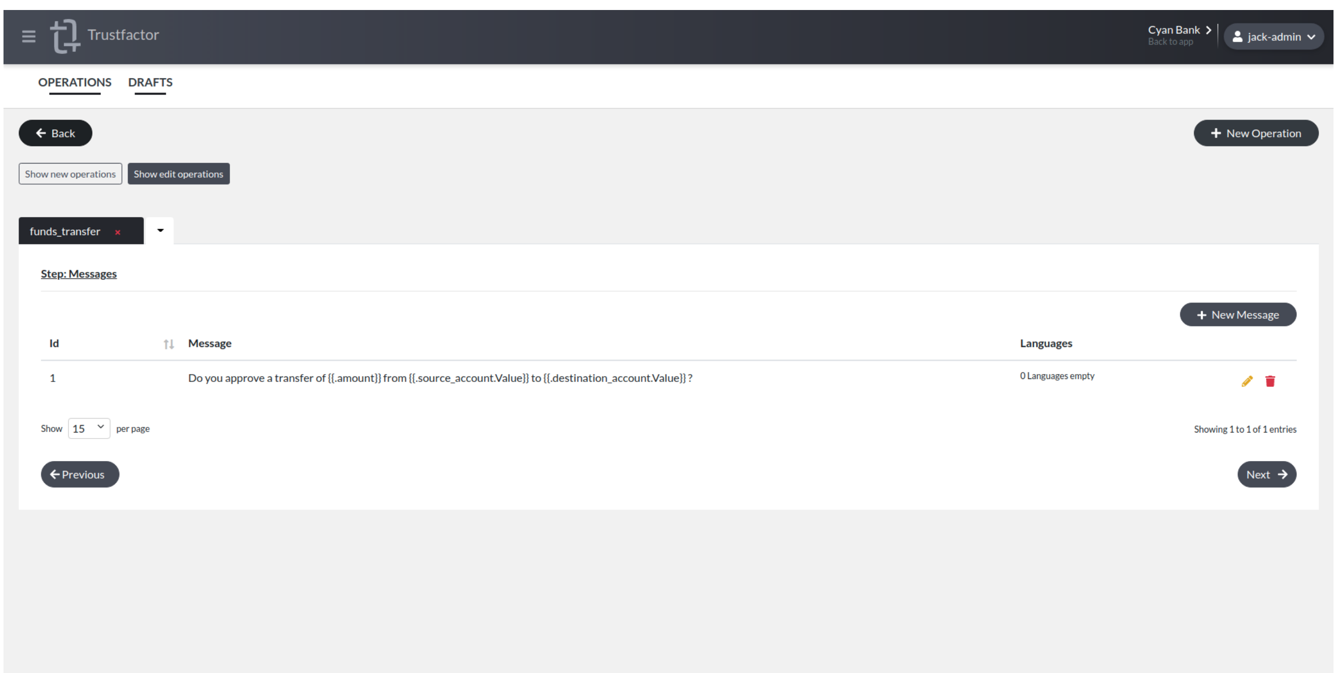
You can also assign risk rules you already created in the Rule Templates tab to the parameter you are creating.



After adding a risk rule, you can define the messages for each risk level, in every supported language you selected in the operation's basic setttings screen.

You can also assign the weight for these parameters, which will be used in the risk calculation for the whole transaction.

Step: Messages The **Messages** screen is where we define the messages shown in the first screen of the decision workflow in the mobile app.



We could add more messages to the operation that can be used in different scenarios by referencing the message ID.

First we define the message for the default language.

After setting the default language message, we must also define the translations for the other languages set in the application's basic settings by pressing *Add languages*. When you are done configuring the message, *0 languages empty* should appear on the messages table.

Referencing Parameter Values in Messages

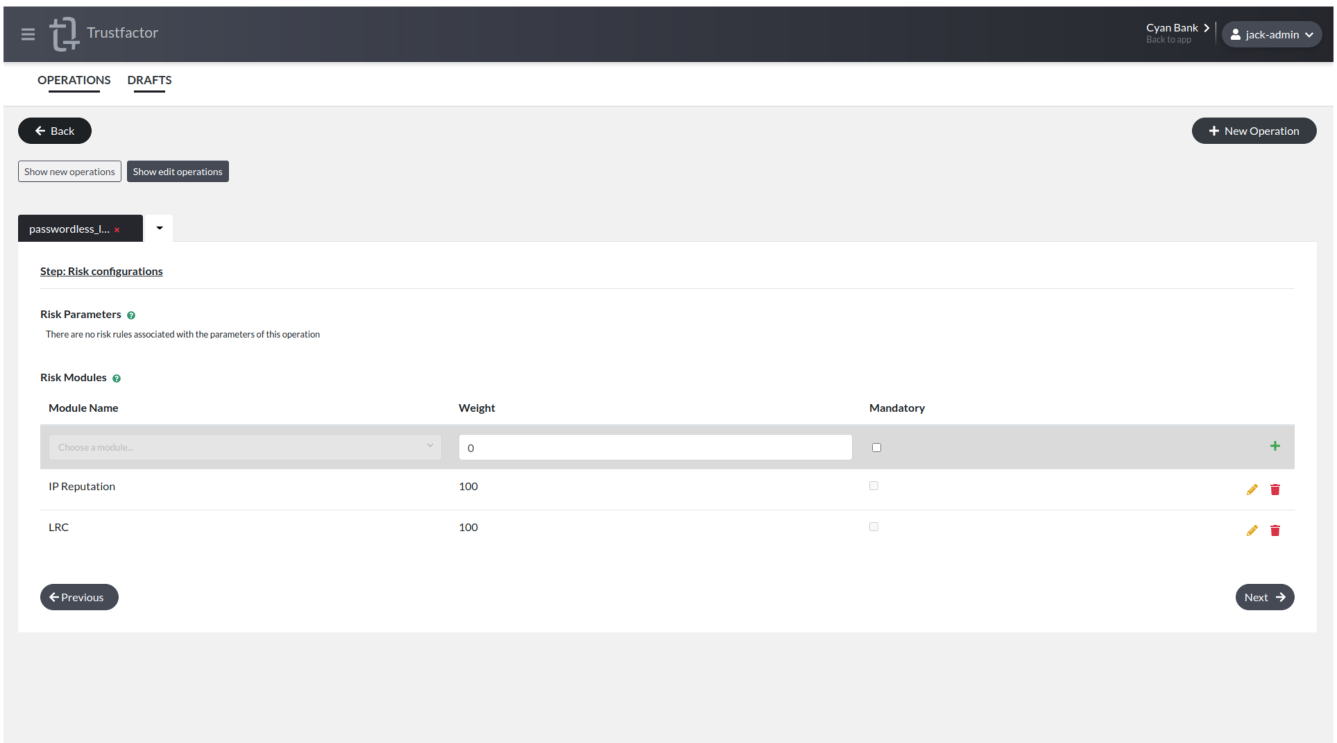
It's possible to reference parameter values in messages by using Go templates. If you have a parameter with the key `username`, you can reference the passed parameter in the message by doing `{{.username.Value}}` . This is straightforward for all parameter types except *money*. For money you may want a formatted string with the amount and the currency and for that you need to use just the parameter key (e.g. if you have a money parameter with key `amount`, then you reference it with `{{.amount}}`).

A full message can be something like:

```
Do you wish to proceed with funds transfer of {{.amount}} ?
```

We can now click *Next ->* in the bottom left corner to proceed.

Step: Risk Settings Here you can configure risk settings for this operation. There are no parameters so there are no settings there. We can however enable Risk Modules.



An operation may define static rules for transaction's risk based on parameters' values. In order to use static rules for a parameter the operation should define:

- The weight of the parameter for the whole transaction's risk evaluation;
- Risk conditions to apply to the parameter for each risk level;
- The expression logic for the parameter for each risk level;
- The risk multiplier for each risk level.

The parameter risk levels will be evaluated from the highest to the lowest. The evaluation stops as soon as a risk level is triggered. The transaction will fail if the parameter's value doesn't fit in any rule.

The transaction's risk level calculation is based on defined thresholds:

- 33 points or less - **Low risk**
- 34 points or more but less than 68 points - **Medium risk**
- 68 points or more - **High risk**

The transaction's risk score is calculated using the following formula:

```
Σ(params risk score) + Σ(modules risk score)
```

The risk score for each parameter is defined by:

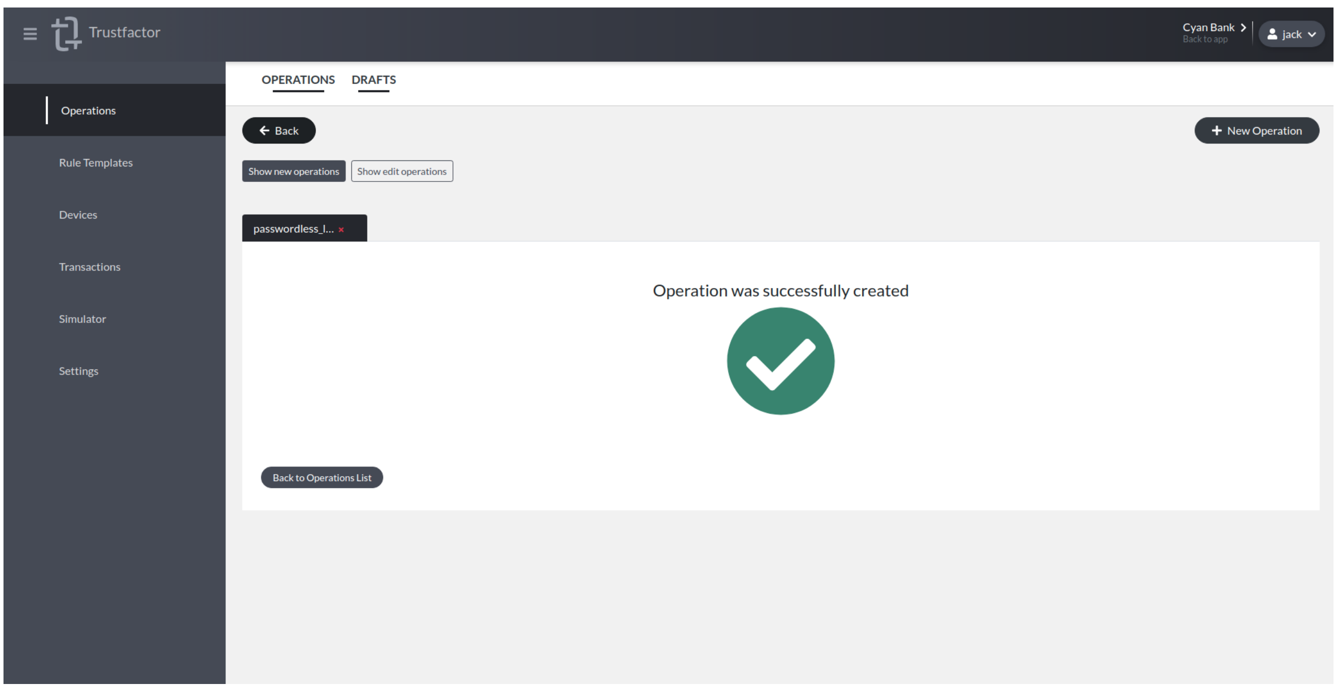
```
Parameter's weight * Parameter's Risk Level Multiplier
```

Risk Modules There are currently two risk modules supported:

- **LRC** - The Location Risk Calculator leverages the geo-location of the user to assess the risk of the transaction. This requires either the client IP (of the application user) who created the transaction to be sent in CreateTransaction metadata or the coordinates (latitude and longitude).
- **IP Reputation** - The IP Reputation risk module uses our threat intelligence sources to flag suspicious source IP addresses. This requires the client IP (of the application user) who created the transaction to be sent in CreateTransaction metadata.

You can select one or both and assign a weight to it. This weight will be factored in the risk calculation for the transaction as defined in the section above. Normally we select both and assign a weight of 100 to each so that if either of these modules are triggered the transaction is bumped to high risk immediately.

Submission When you are ready and get no errors on the validation step, you can then create the operation.



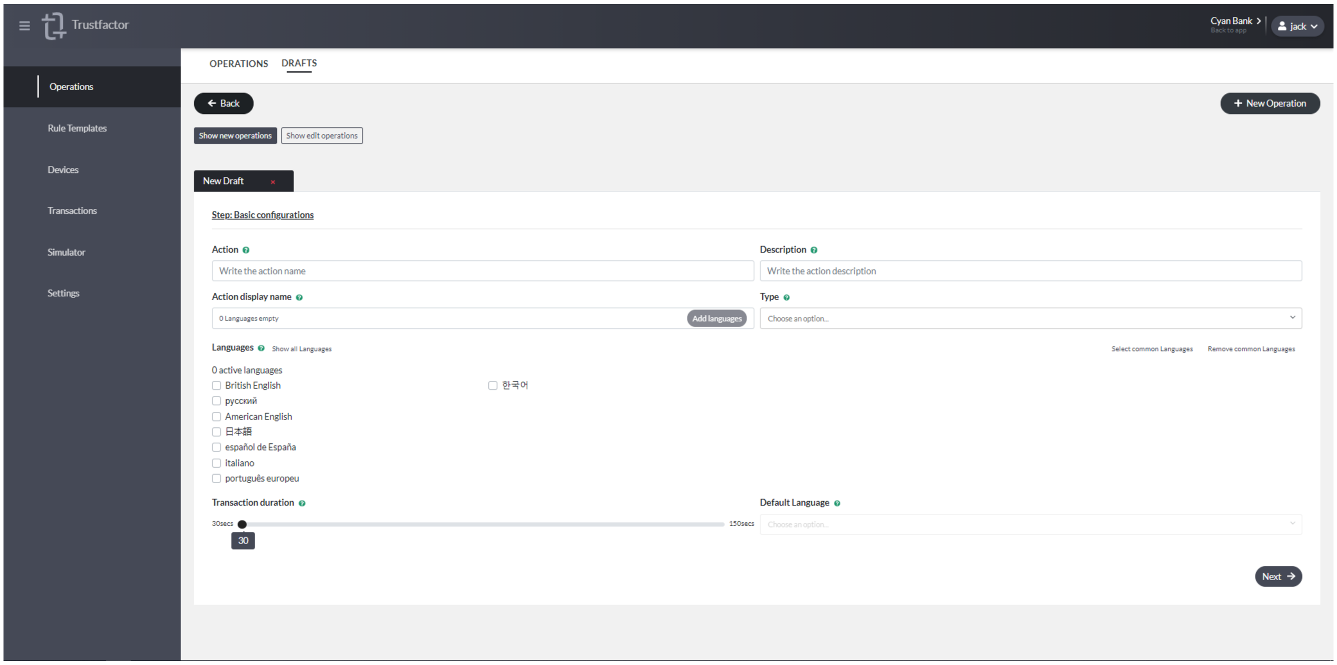
After creating the operation, you can use the Simulator to check what the authentication request will look like for a given user.

Example 1 - Password-less Login

In order to explore operations usage, we will set a few common scenarios that can leverage TrustFactor operations for multi-factor authentication.

One of the most common use cases for TrustFactor is password-less login. This is one of the simplest operations to configure on TrustFactor, as it usually does not have any parameters, just a simple message like "You have requested to log in on using . Do you wish to proceed?".

The first thing we need to do is access the Operations menu in our application and press *+ New Operation*. This opens the basic configurations screen for operations as shown below.



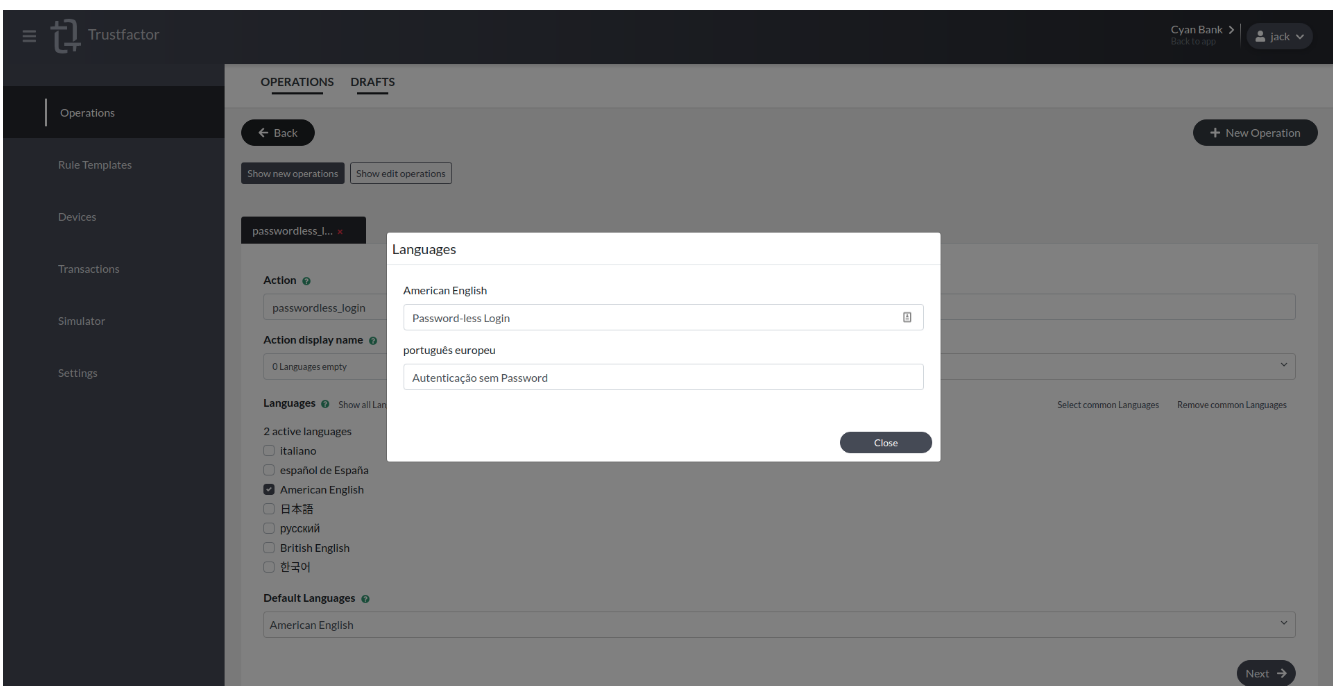
Step: Basic Settings screen For the password-less login example, let's configure two languages: **Portuguese** and **American English**.

- **Action**

For this example, we'll just set it to *passwordless_login*.

- **Action display name**

We set "Password-less Login" for the American English language and "Autenticação sem Password" for the Portuguese language.



- **Description**

A password-less login operation example.

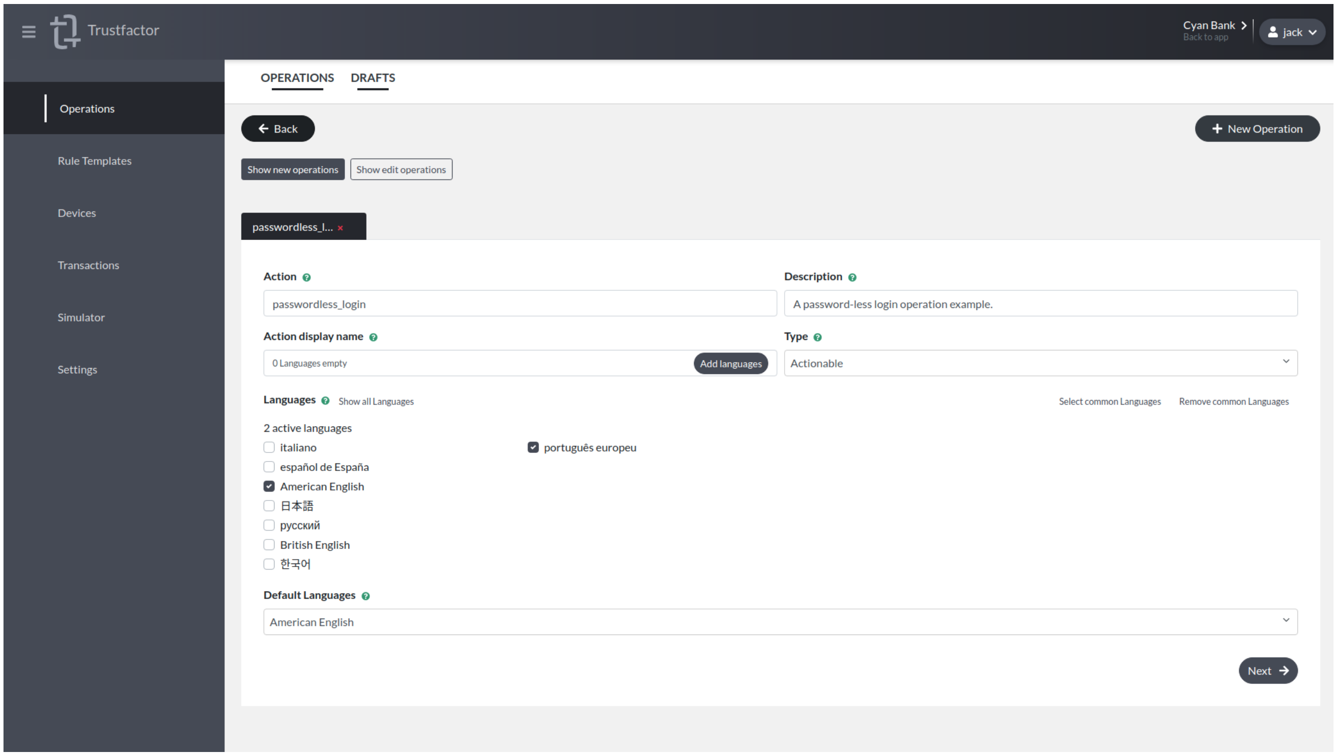
- **Type**

For the password-less login example, this must be set to **Actionable**, because we want to know if the user approved or rejected.

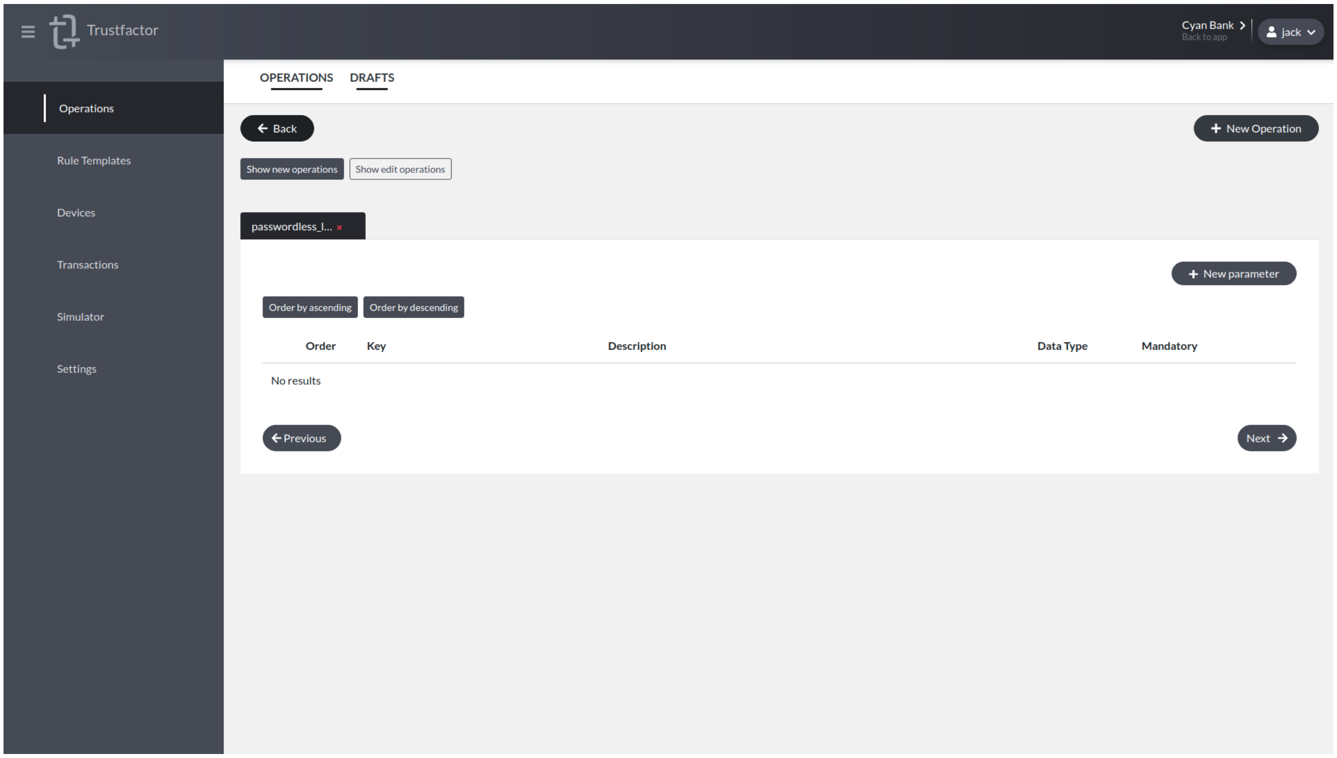
- **Default Language**

We'll set the default language to American English.

The basic configurations screen for password-less login should look like the screenshot below. We can now proceed to the next step by pressing the *Next* -> button in the bottom left corner.



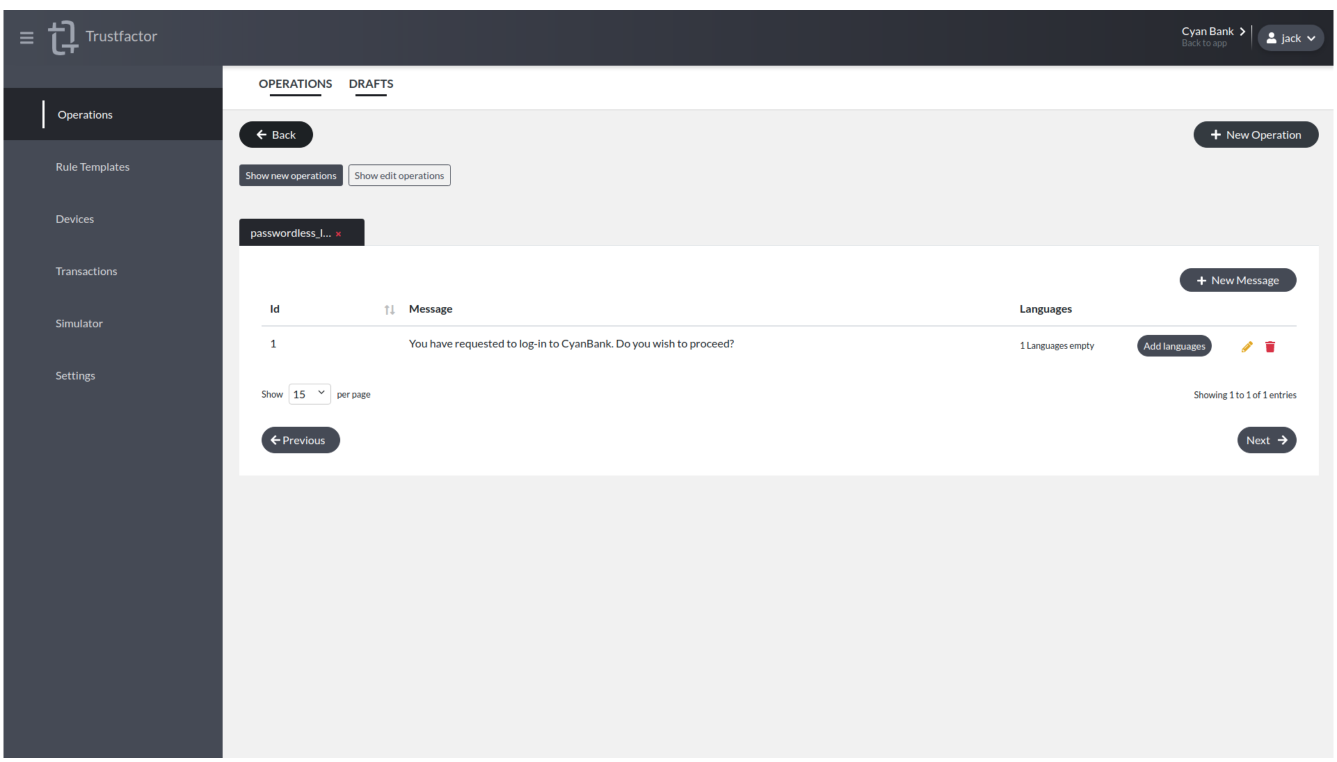
Step: Parameters For this example, we will not define any parameter, so we just press the *Next* -> button.



Step: Messages For the password-less login example, we must define one message for each language defined in the operation's basic settings.

We could more messages to the operation that can be used in different scenarios by referencing the message ID.

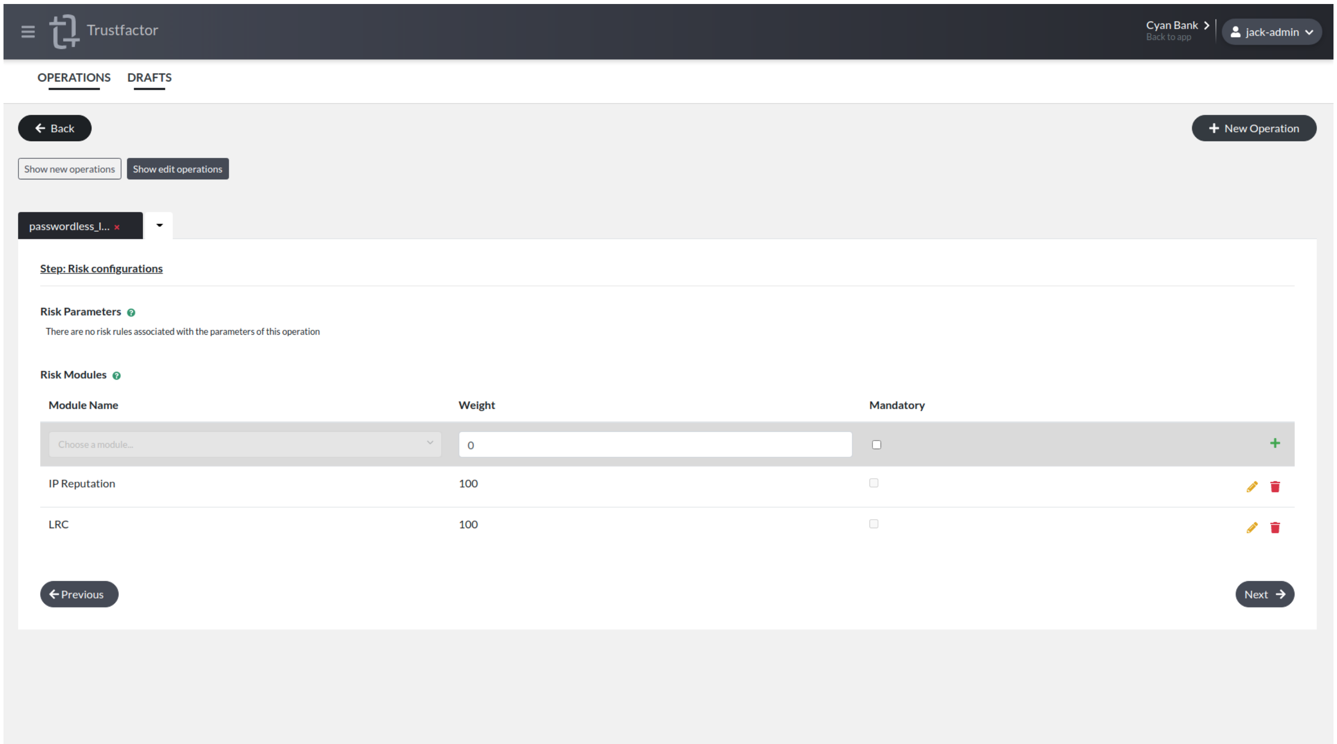
First we define the message for the default language.



After setting the default language message (American English), we must also define the translations for the other languages set in the application's basic settings by pressing *Add languages*. When you are done configuring the message, *0 languages empty* should appear on the messages table.

We can now click *Next ->* in the bottom left corner to proceed.

Step: Risk Settings Here you can configure risk settings for this operation. There are no parameters so there are no settings there. We can however enable Risk Modules.



An operation may define static rules for transaction's risk based on parameters' values. In order to use static rules for a parameter the operation should define:

- The weight of the parameter for the whole transaction's risk evaluation;
- Risk conditions to apply to the parameter for each risk level;
- The expression logic for the parameter for each risk level;
- The risk multiplier for each risk level.

The parameter risk levels will be evaluated from the highest to the lowest. The evaluation stops as soon as a risk level is triggered. The transaction will fail if the parameter's value doesn't fit in any rule.

The transaction's risk level calculation is based on defined thresholds:

- 33 points or less - **Low risk**
- 34 points or more but less than 68 points - **Medium risk**
- 68 points or more - **High risk**

The transaction's risk score is calculated using the following formula:

`Σ(params risk score) + Σ(modules risk score)`

The risk score for each parameter is defined by:

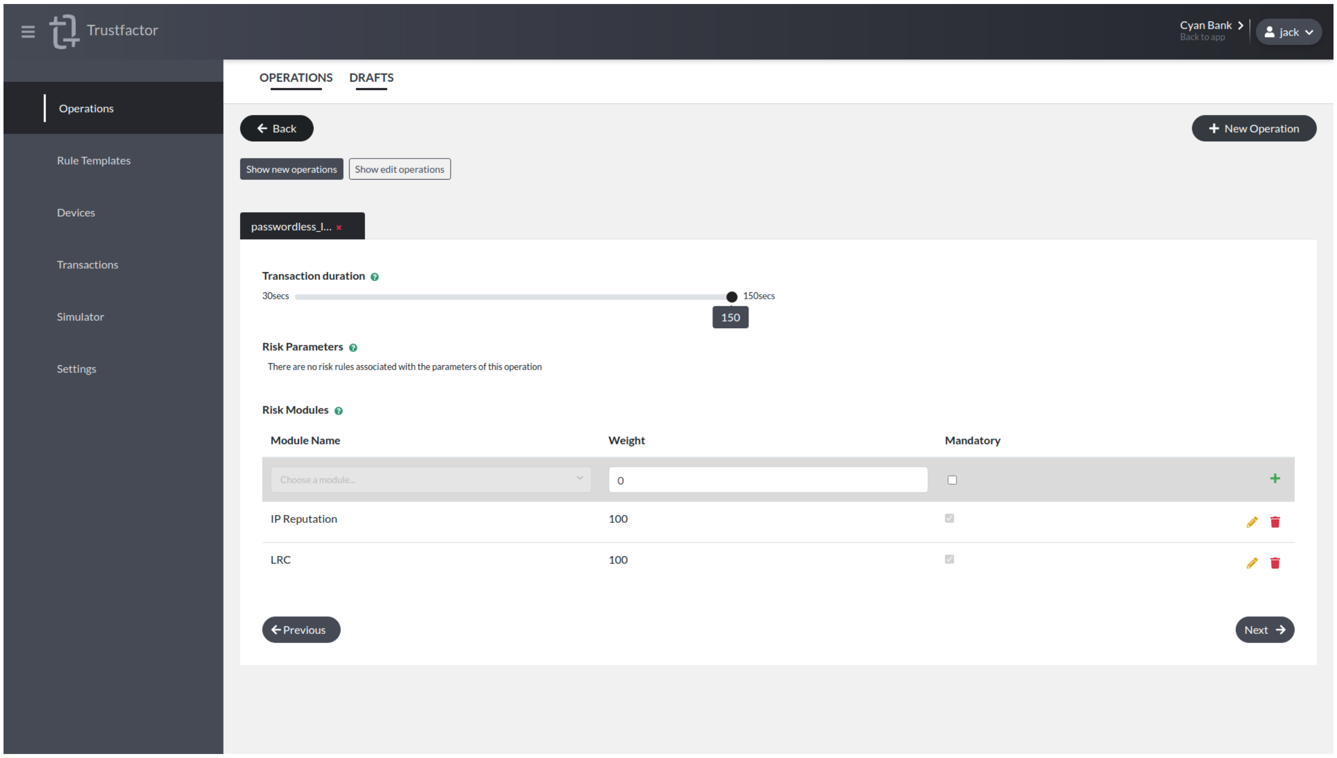
`Parameter's weight * Parameter's Risk Level Multiplier`

Risk Modules There are currently two risk modules supported:

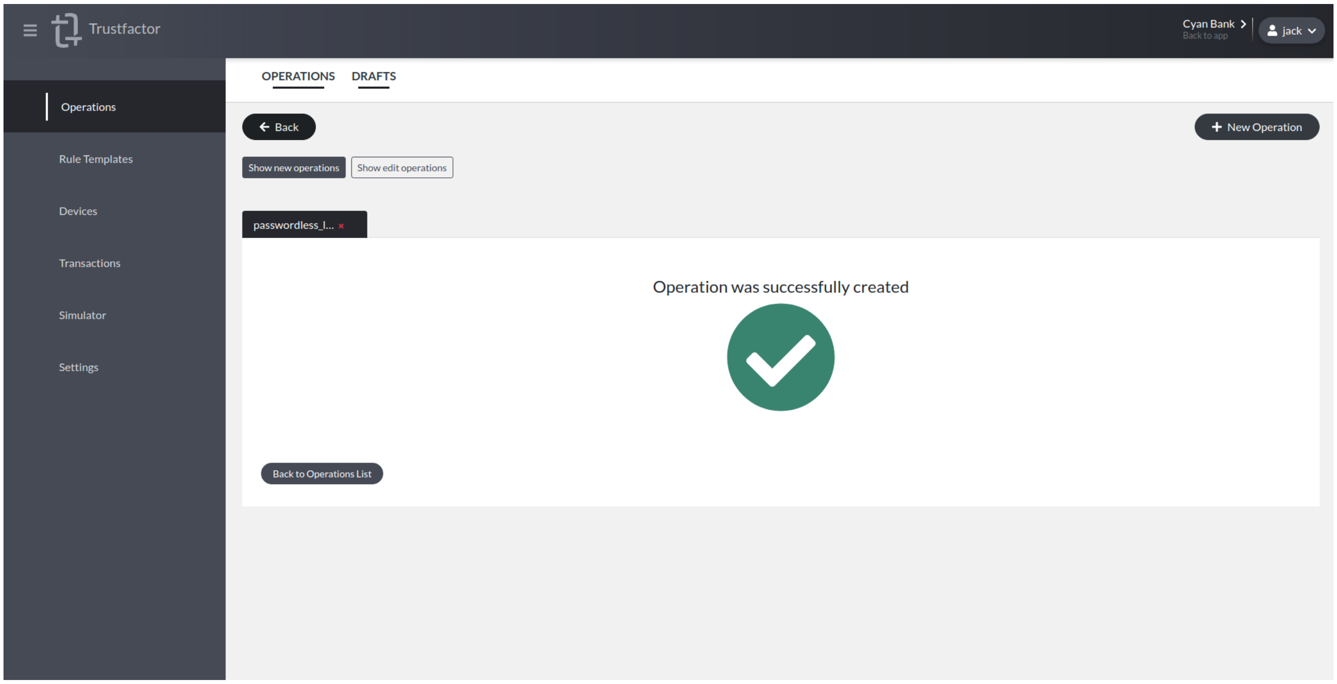
- **LRC** - The Location Risk Calculator leverages the geo-location of the user to assess the risk of the transaction. This requires either the client IP (of the application user) who created the transaction to be sent in CreateTransaction metadata or the coordinates (latitude and longitude).

- **IP Reputation -** The IP Reputation risk module uses our threat intelligence sources to flag suspicious source IP addresses. This requires the client IP (of the application user) who created the transaction to be sent in CreateTransaction metadata.

You can select one or both and assign a weight to it. This weight will be factored in the risk calculation for the transaction as defined in the section above. Normally we select both and assign a weight of 100 to each so that if either of these modules are triggered the transaction is bumped to high risk immediately.



Submission When you are ready and get no errors on the validation step, you can then create the operation.



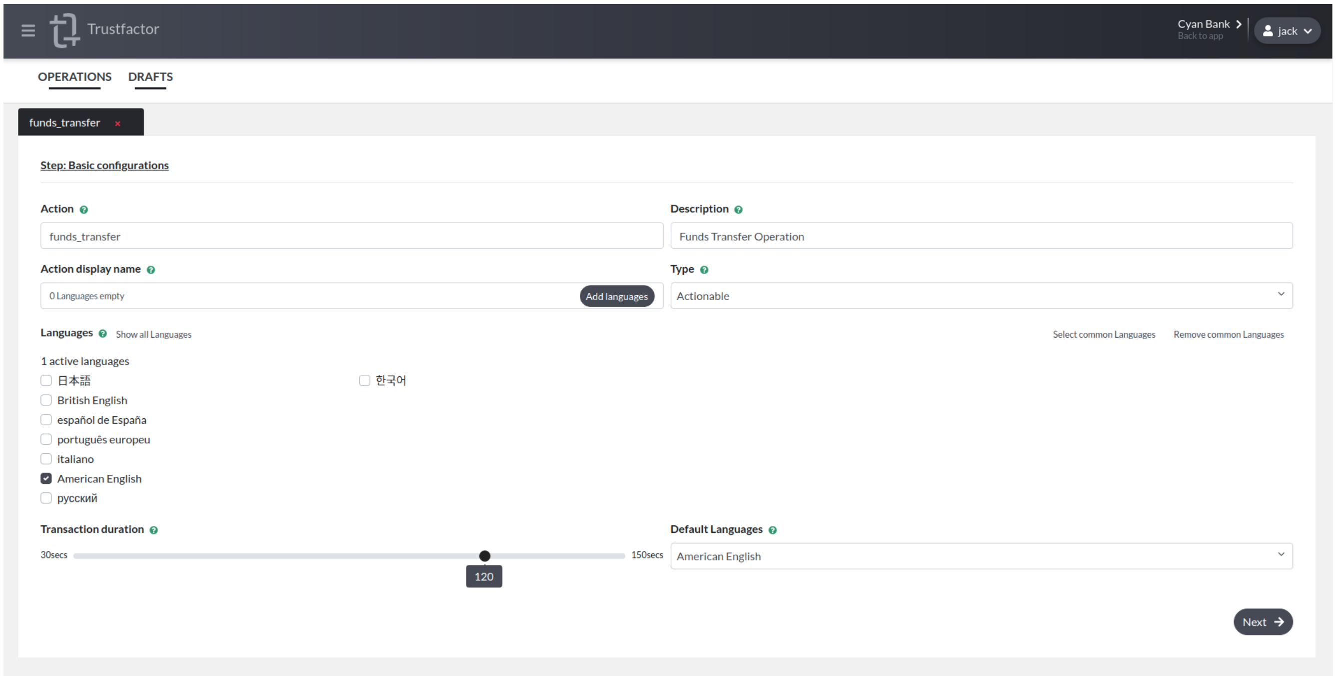
After creating the operation, you can use the Simulator to check what the authentication request will look like for a given user.

Example 2 - Funds Transfer

In this example we will go through the operation create workflow but now we will define parameters and assign risk rules to them. We will take the previous example defined in Example - Funds Transfer.

Let's go through the operations creation workflow.

Step: Basic Configuration We define the action, languages, description and proceed to the parameters screen.



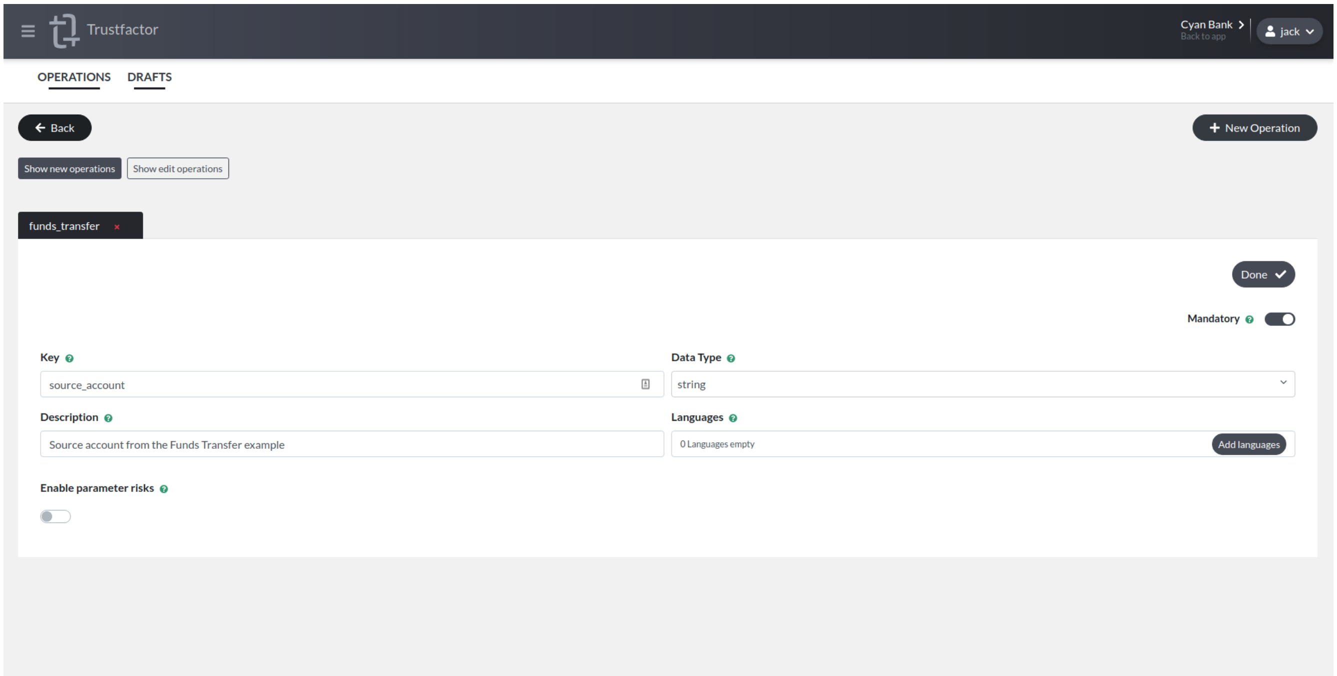
Step: Parameters In order to create the Funds Transfer operation, we're going to need to define 3 parameters:

- Source Account
- Destination Account
- Transaction Amount

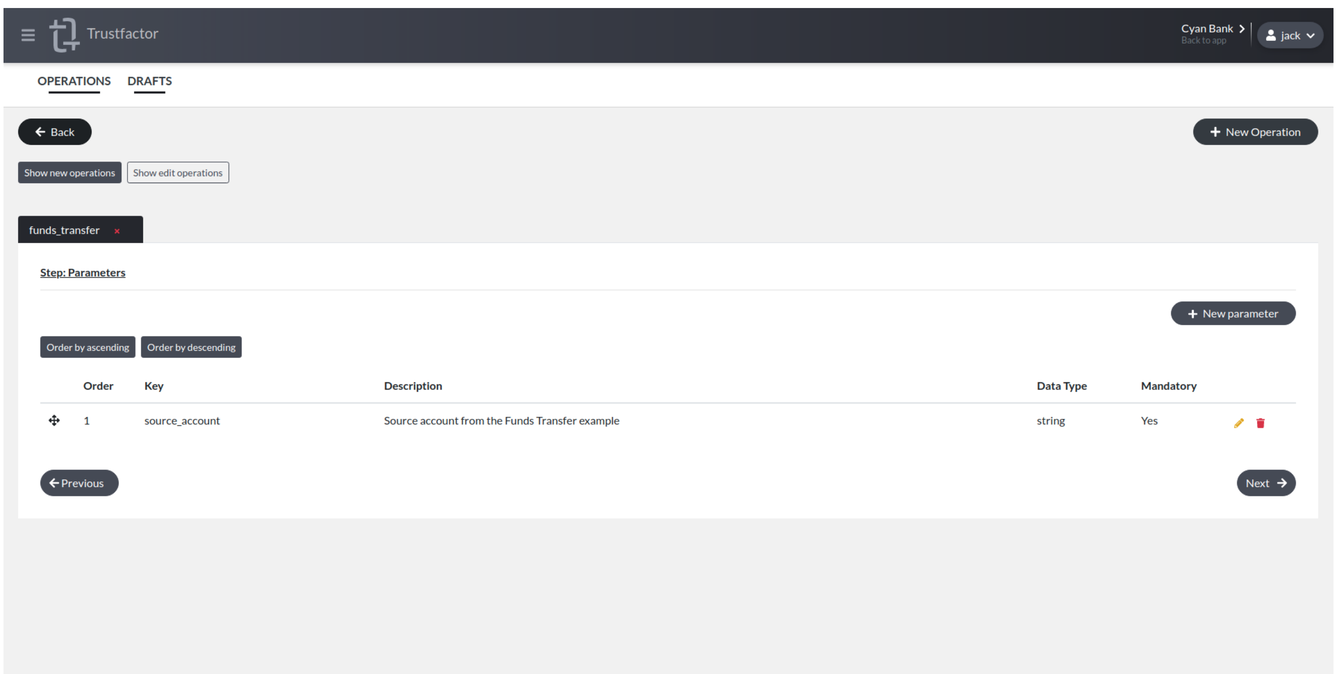
In the parameters screen, choose *+ New Parameter* . This will take you to the parameter creation screen. The first parameter we create is going to define is the source account. Let's fill in the required parameter fields.

- **Key** - `source_account`
- **Data Type** - `string`
- **Description** - `Source account from the Funds Transfer example`
- **Default Language Parameter Label (english)** - `Source Account`

This parameter does not have any risk rules associated with it, so we leave the *Enable parameter risk rules* toggle disabled for this one.



Now we can press *Done* to save the parameter. A new line should appear on the *Step: Parameters* screen as shown below.



Now we go ahead and create the other two parameters. **Destination Account** is next and it's definition is very similar to **Source Account**:

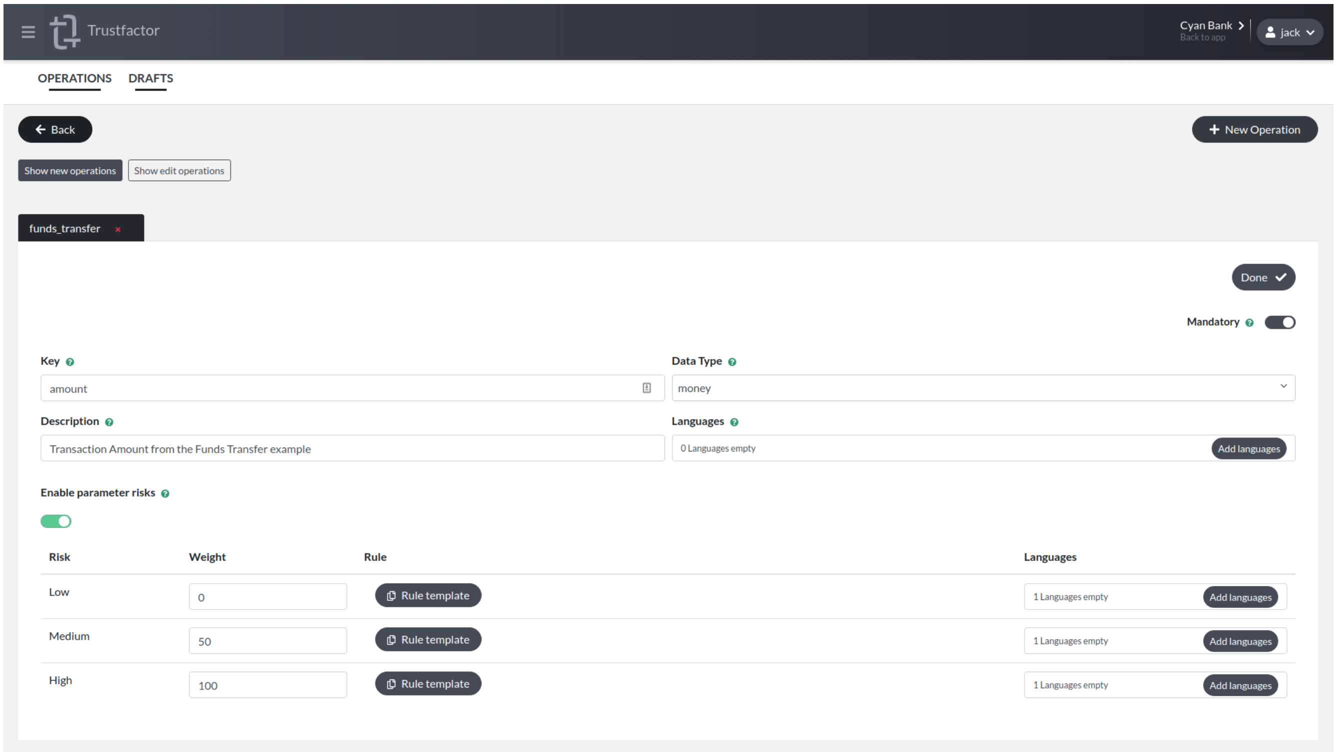
- **Key** - `destination_account`
- **Data Type** - `string`
- **Description** - `Destination account from the Funds Transfer example`
- **Default Language Parameter Label (english)** - `Destination Account`

Transaction Amount is the most interesting parameter to create, because it needs risk rules. First we define the required parameters:

- **Key** - `amount`

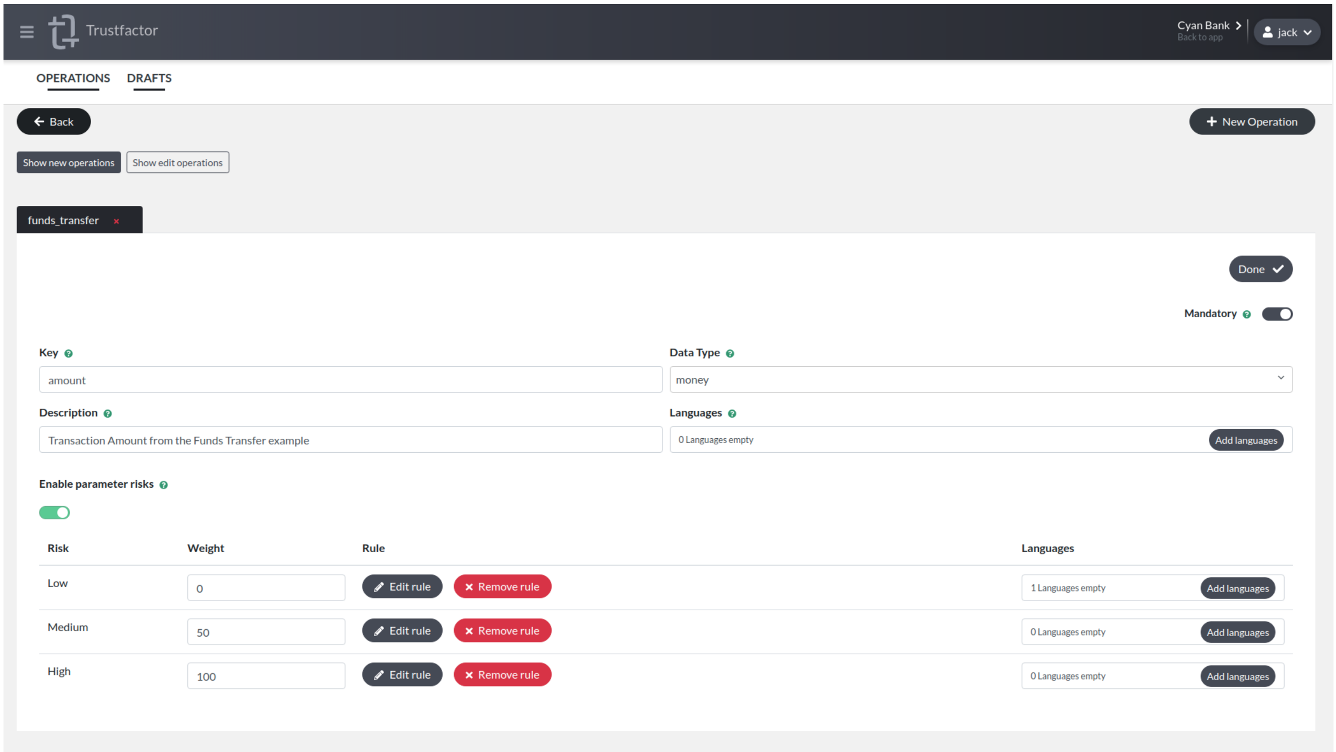
- **Data Type** - `money`
- **Description** - `Transaction Amount from the Funds Transfer example`
- **Default Language Parameter Label (english)** - `Transaction Amount`

Now we need to press *Enable parameter risk rules* to use the rule templates we created earlier on this operation. When you press the button, you should see a screen as shown below:

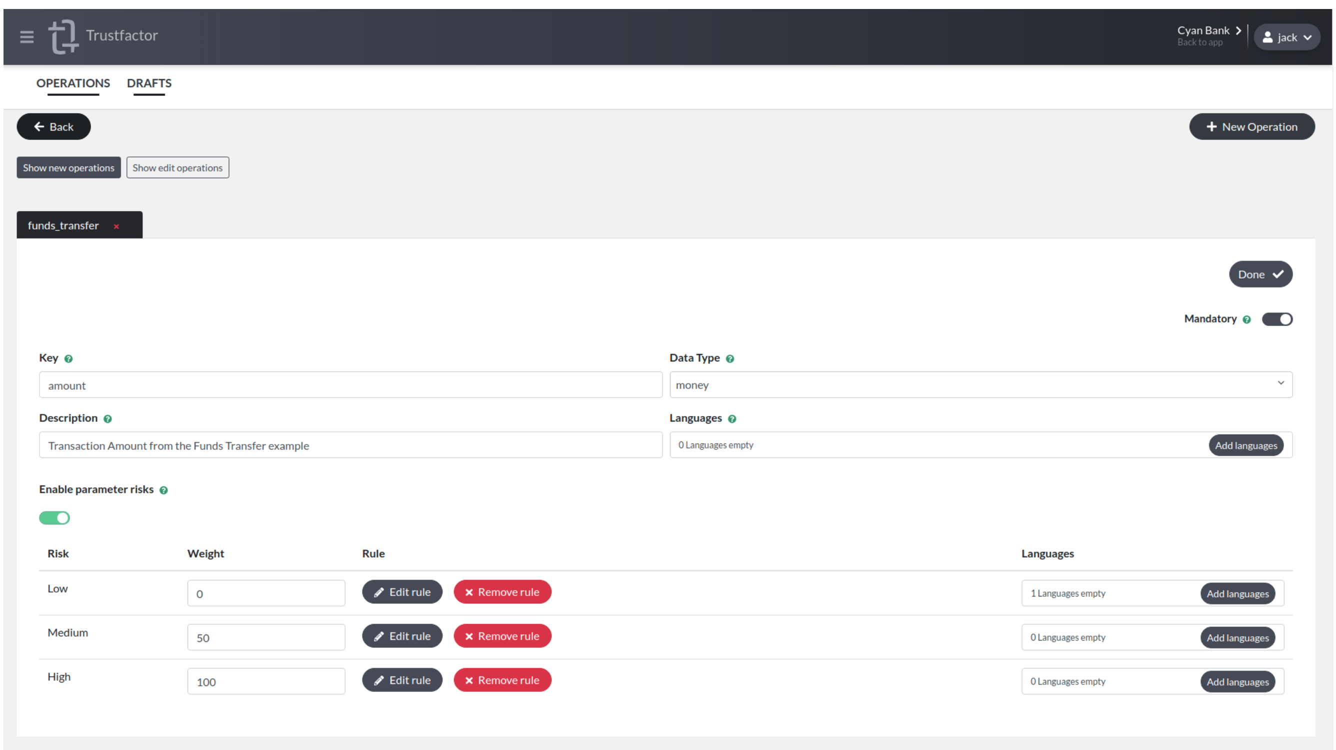


We need to configure the rules that trigger each risk rating (low, medium and high) as explained in the Risk Settings Section. For *low* risk, we're going to use the rule `funds_transfer_low_risk` that was created earlier. When *low* risk transactions are created, we don't want to show any risk message above the parameter, but we do for *medium* and *high* risk so let's configure those by setting the *rule template* and then pressing the *Add languages* button to add the risk message to be shown. For medium risk, let's add the message `Please validate the transaction amount before proceeding`. For high risk, we add `Attention: High Transaction Amount! Please validate before proceeding`.

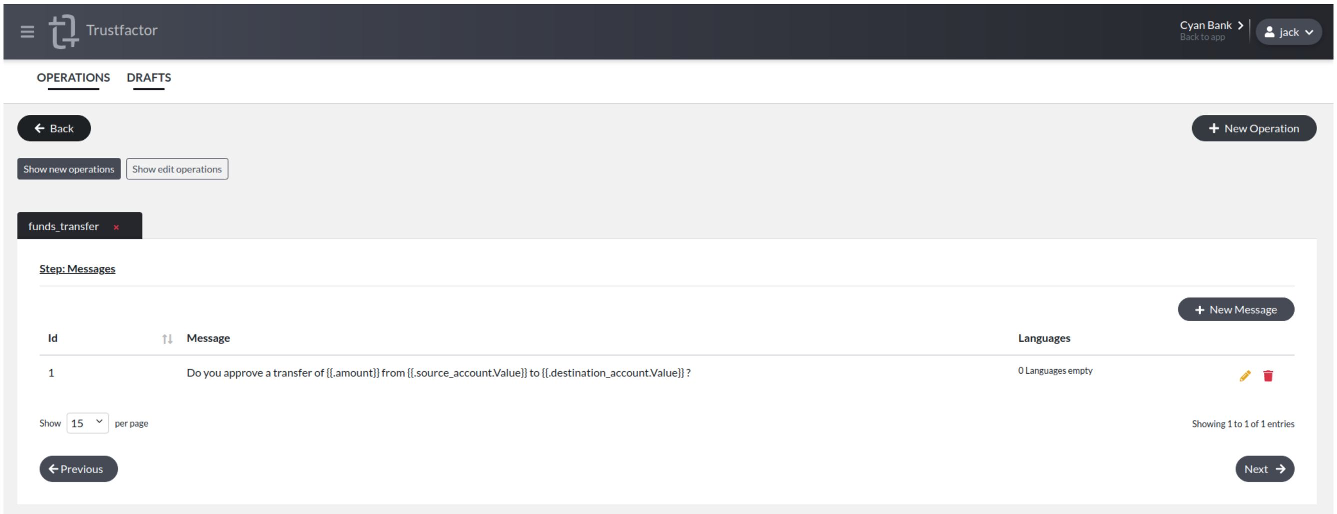
When you are done the screen should look like the screenshot below.



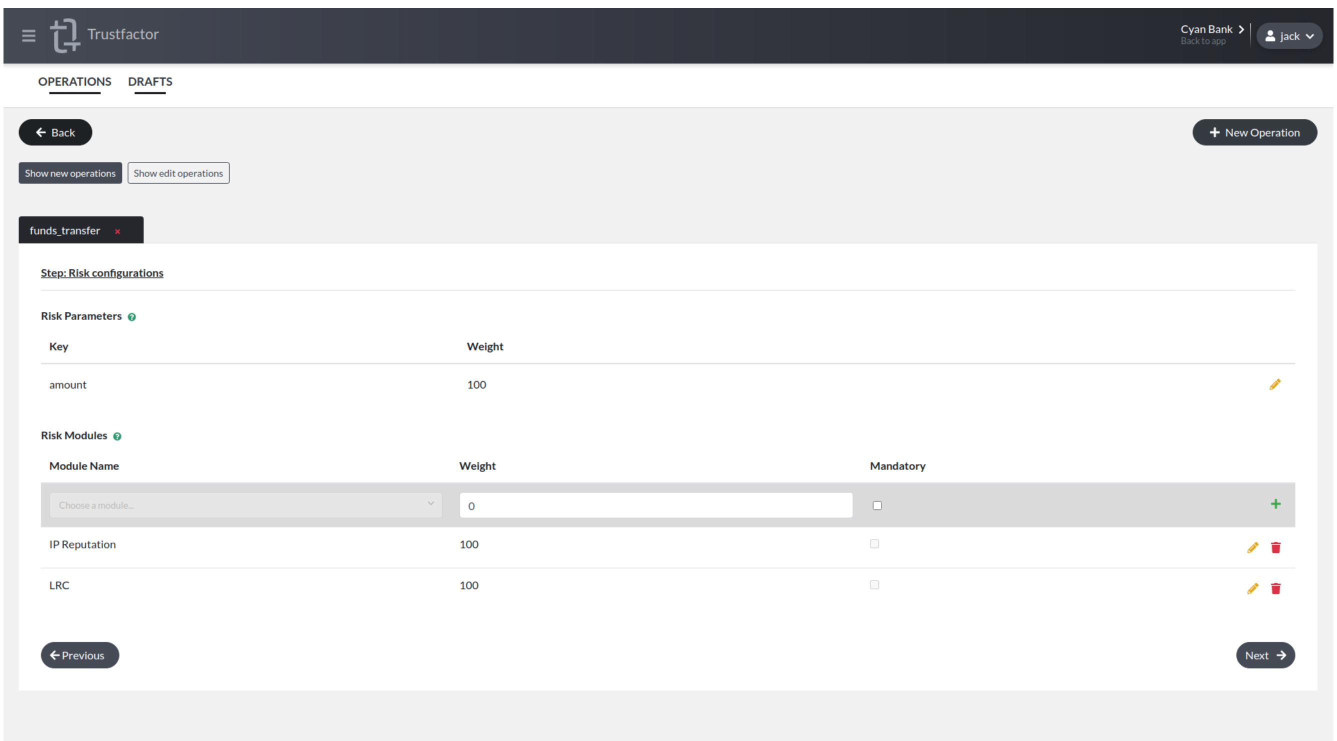
After pressing *Done*, you will see all of the parameters defined as shown in the screenshot below.



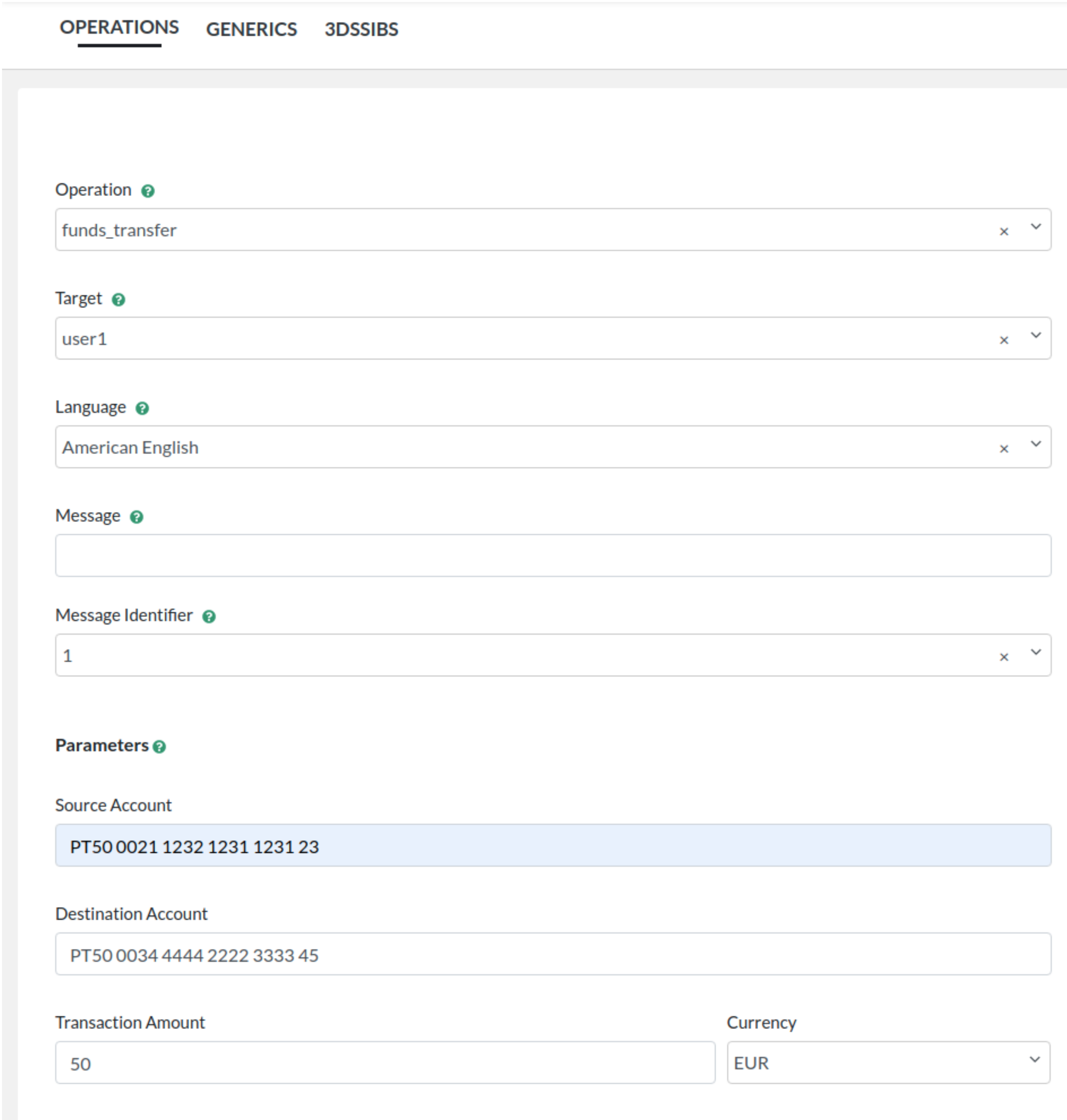
Step: Messages In the messages screen, we can reference the parameter values in order to include them in the message using Go templating as shown below. Later we will reference this Message Identifier (1) in the simulator to see what will be shown to the end user.



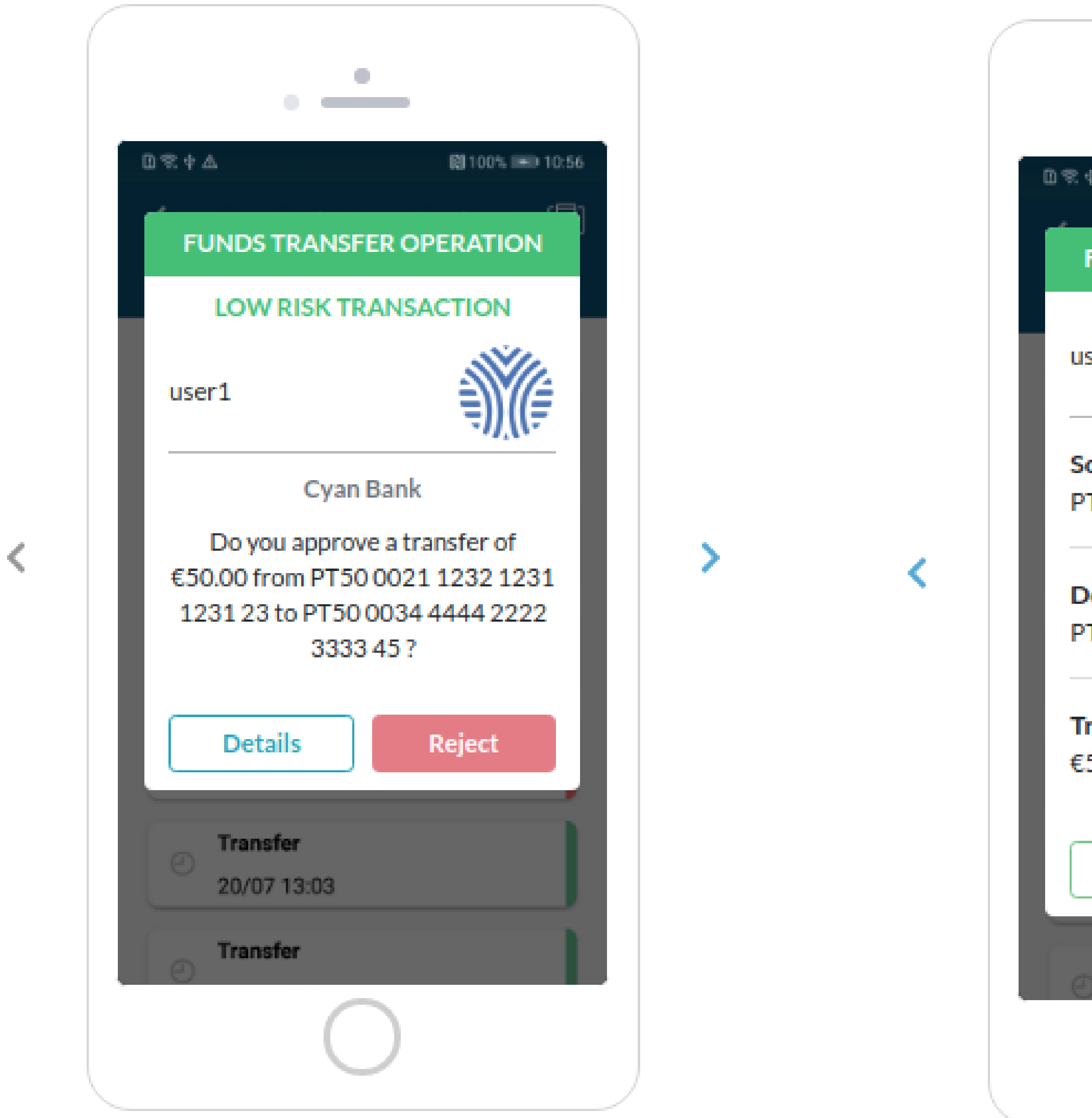
Step: Risk Settings In the risk configurations screen, we set the weight of the Amount parameter to `100`. This means that if the amount parameter is flagged as high risk, the whole transaction is rated high risk. If we lower this weight we may create scenarios where a high risk parameter only causes the transaction to be rated medium risk, but this is not what we want in this case. We also add LRC and IP Reputation risk modules with weight set to `100` because if either module flags the transaction we want it to be high risk as well.



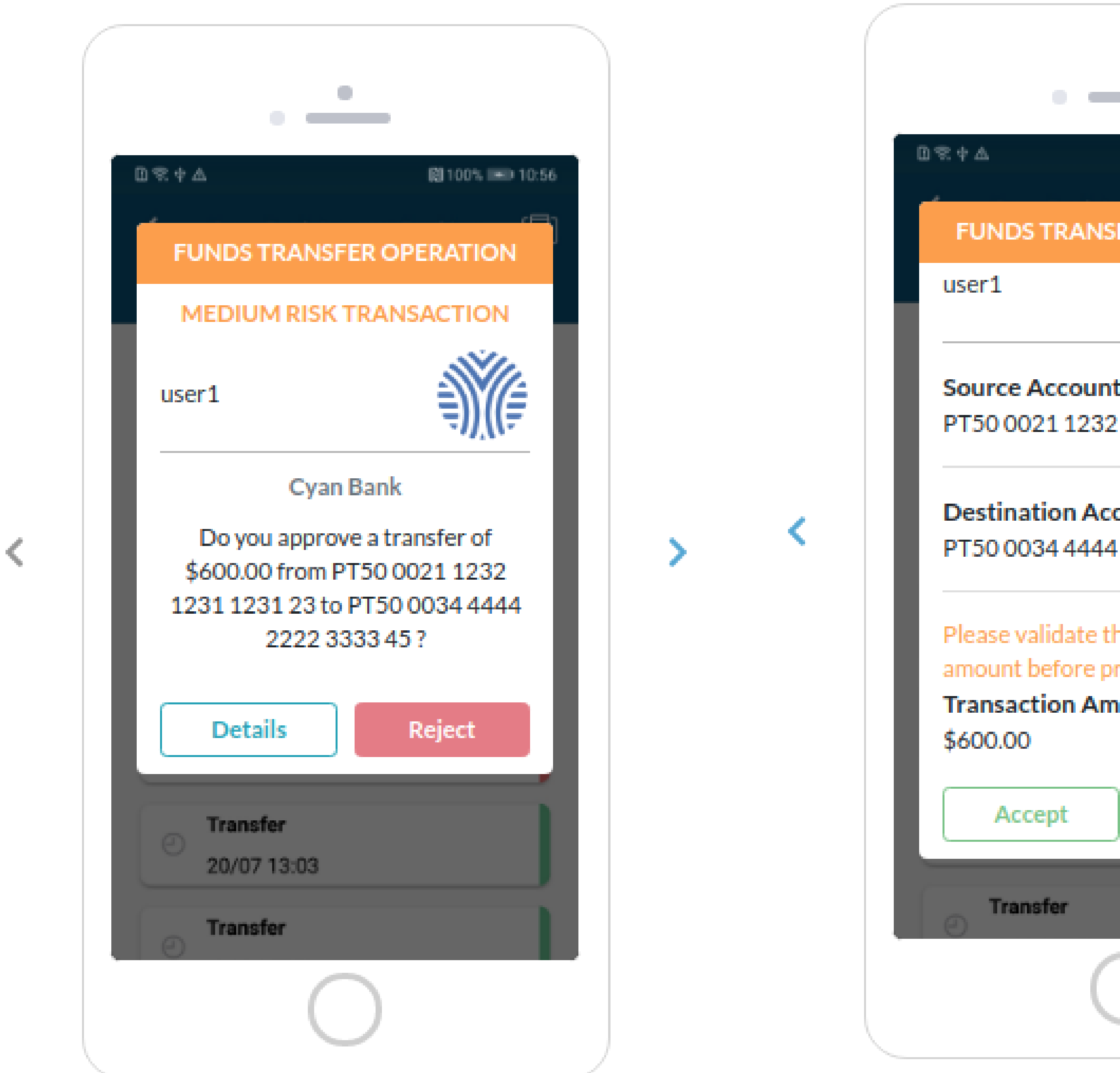
Simulator If we use the Simulator on the newly created transaction, we can see how it will be shown to the end user. We select the `funds_transfer` operation and now the form includes the parameters we need to fill out. We pick a user and set the amount to 50 EUR.



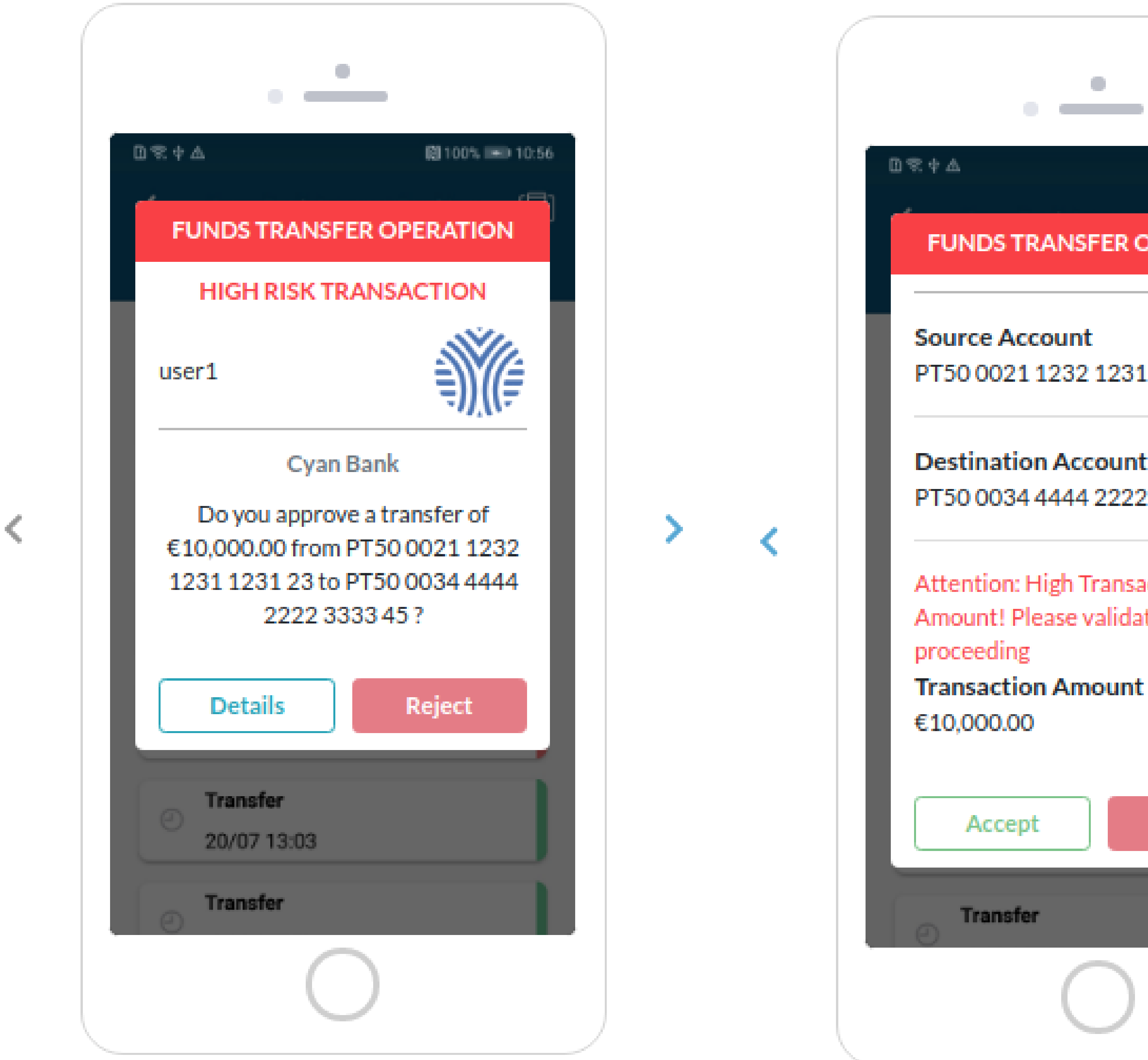
When we press *Simulate* at the bottom, we will see both screens render as they would be shown to that user:



This shows no risk messages whatsoever above the parameters. Now let's try to trigger a medium risk transaction by setting the amount to US $600 .



The medium risk message is now shown above the Transaction Amount parameter as we defined it. Now let's try 10.000 EUR to trigger the high risk rule.

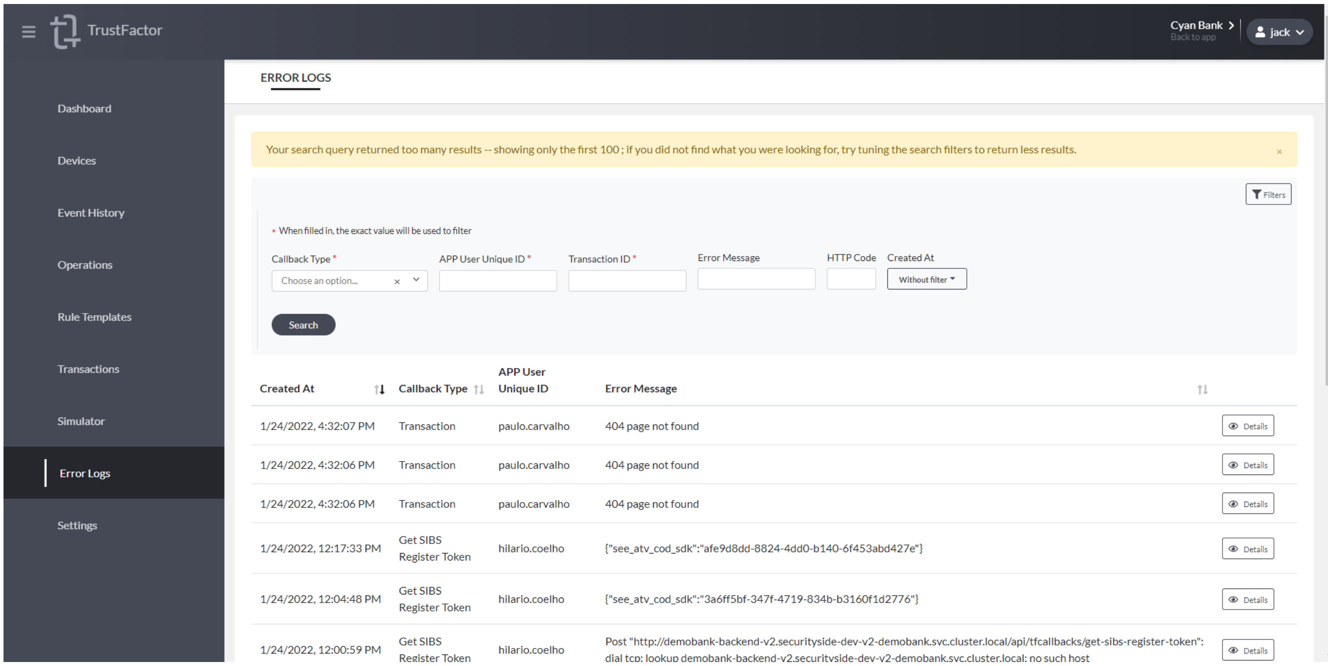


This shows use the High Risk rule has been triggered and the message is shown accordingly.

Error Logs

Error logs are used to diagnose failed TrustFactor callbacks.

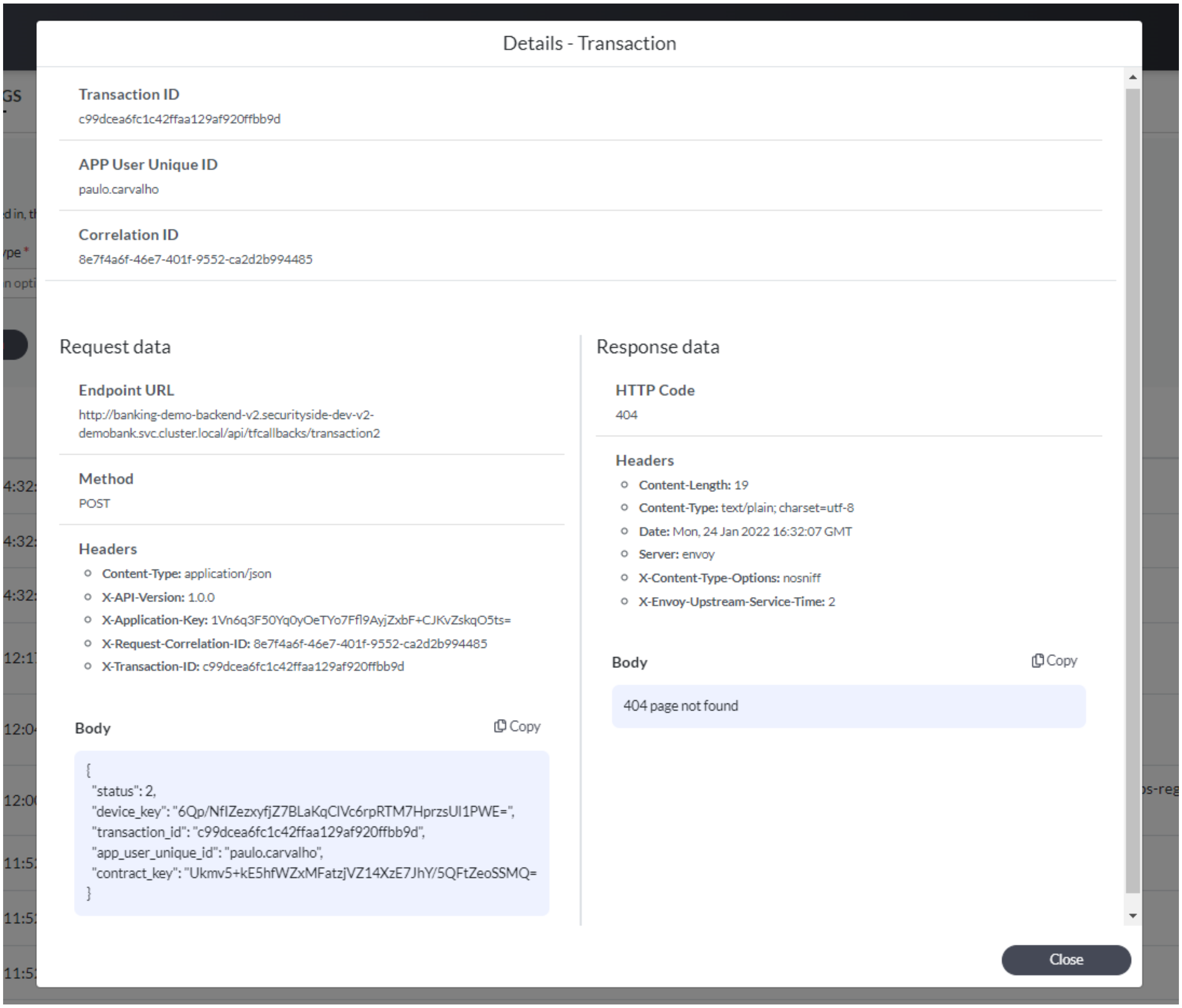
Permissions required to view this screen: - **View Logs**



You can filter for:

- Callback Type
- App User Unique ID
- Transaction ID
- Error Message
- HTTP Code
- Created At (time period)

By pressing the "Details" button at the end of the table row, you will be able to see the error details.



Details - Transaction
Transaction ID
c99dcea6fc1c42ffaa129af920ffbb9d
APP User Unique ID
paulo.carvalho
Correlation ID
8e7f4a6f-46e7-401f-9552-ca2d2b994485
Request data
Endpoint URL
http://banking-demo-backend-v2.securityside-dev-v2-demobank.svc.cluster.local/api/tfcallbacks/transaction2
Method
POST
Headers
Content-Type: application/json
X-API-Version: 1.0.0
X-Application-Key: 1Vn6q3F50Yq0yOeTYo7Ffl9AyjZxbF+CJKvZskqO5ts=
X-Request-Correlation-ID: 8e7f4a6f-46e7-401f-9552-ca2d2b994485
X-Transaction-ID: c99dcea6fc1c42ffaa129af920ffbb9d
Body
Copy
{
"status": 2,
"device_key": "6Qp/NfIZezxyfjZ7BLaKqClVc6rpRTM7HprzsUI1PWE=",
"transaction_id": "c99dcea6fc1c42ffaa129af920ffbb9d",
"app_user_unique_id": "paulo.carvalho",
"contract_key": "Ukmv5+kE5hfWZxMFatzjVZ14XzE7JhY/5QFtZeoSSMQ=
}
Response data
HTTP Code
404
Headers
Content-Length: 19
Content-Type: text/plain; charset=utf-8
Date: Mon, 24 Jan 2022 16:32:07 GMT
Server: envoy
X-Content-Type-Options: nosniff
X-Envoy-Upstream-Service-Time: 2
Body
Copy
404 page not found
Close

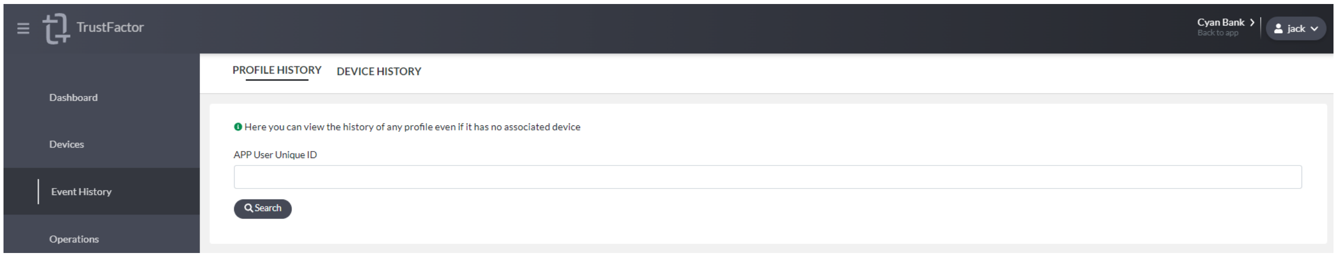
Event History

Profile History

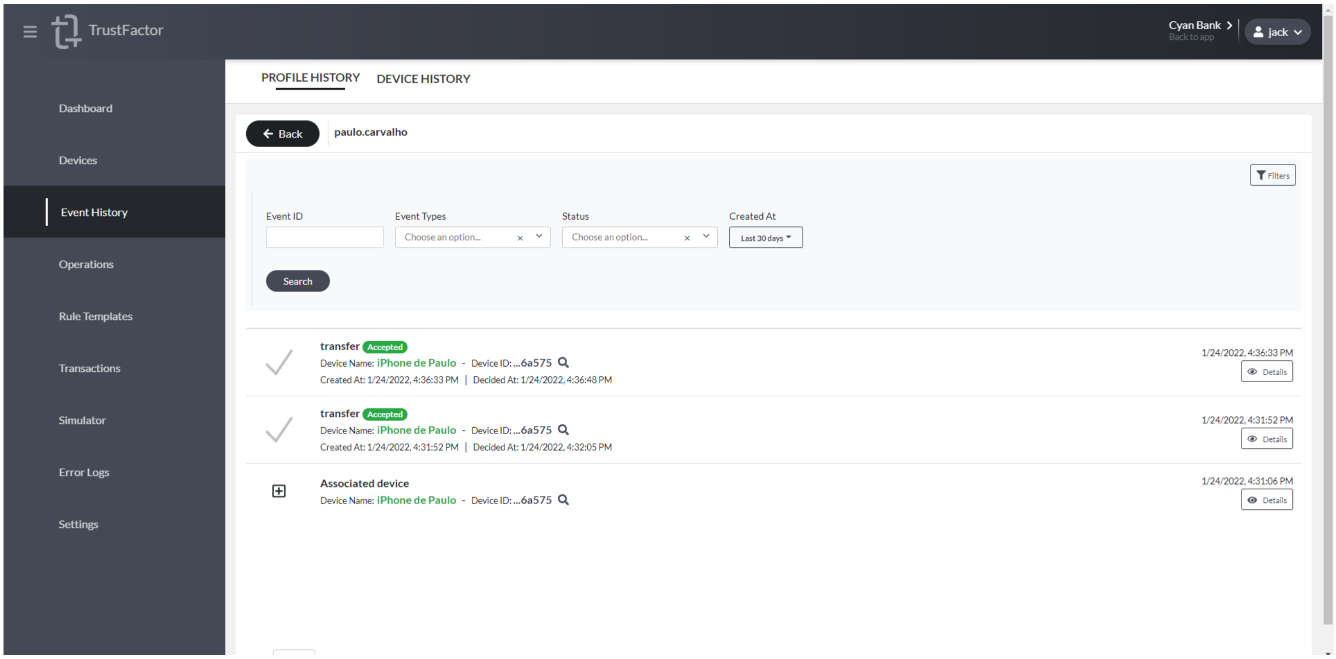
The profile history is used to consult all the activity performed by a profile.

Permissions required to view this screen: **- View Contracts - View Transactions**

Here you can search by device.



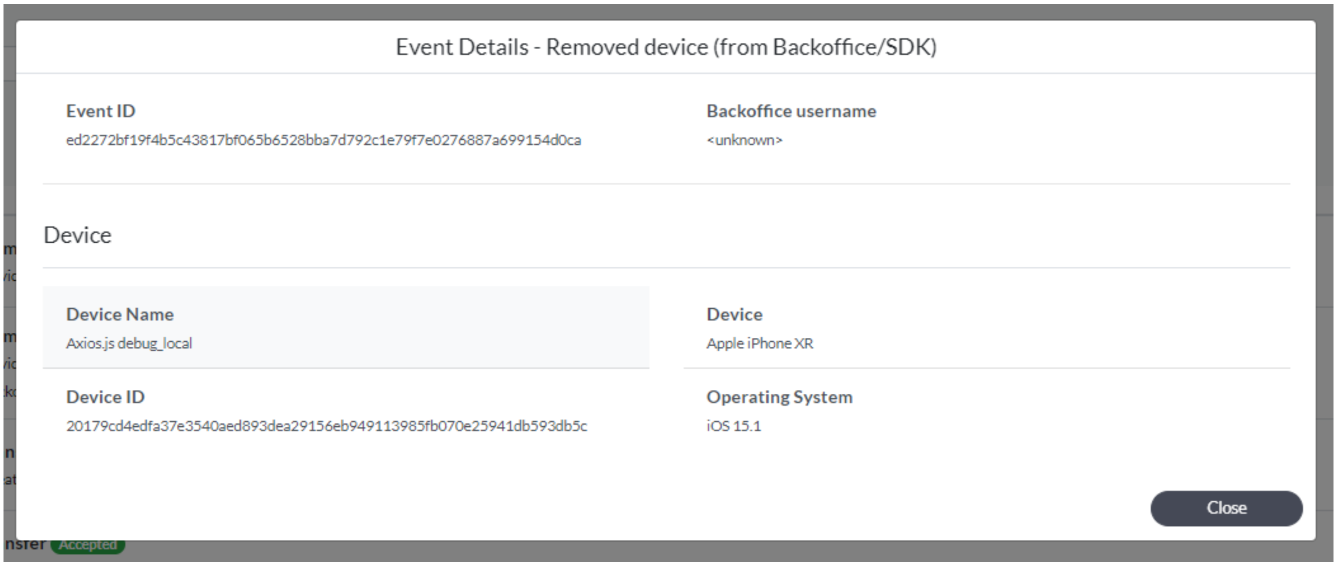
Here you can see profile event list.



You can filter for:

- Event ID
- Event Types
- Status
- Created At (time period)

By pressing the "Details" button at the end of the table row, you will be able to see the event details.

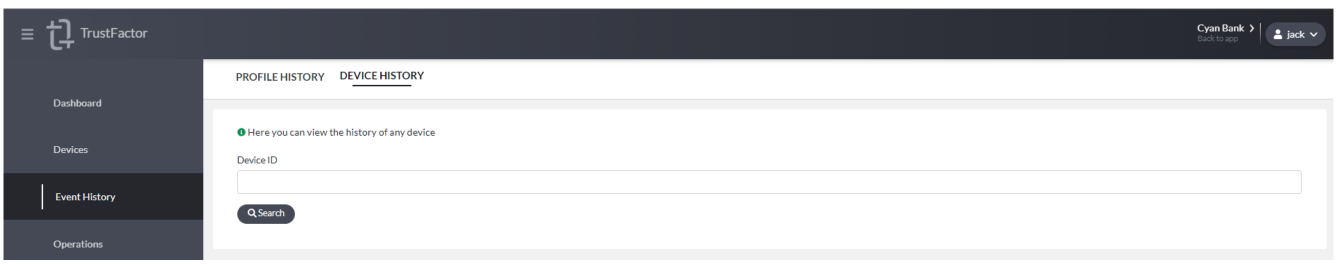


Device History

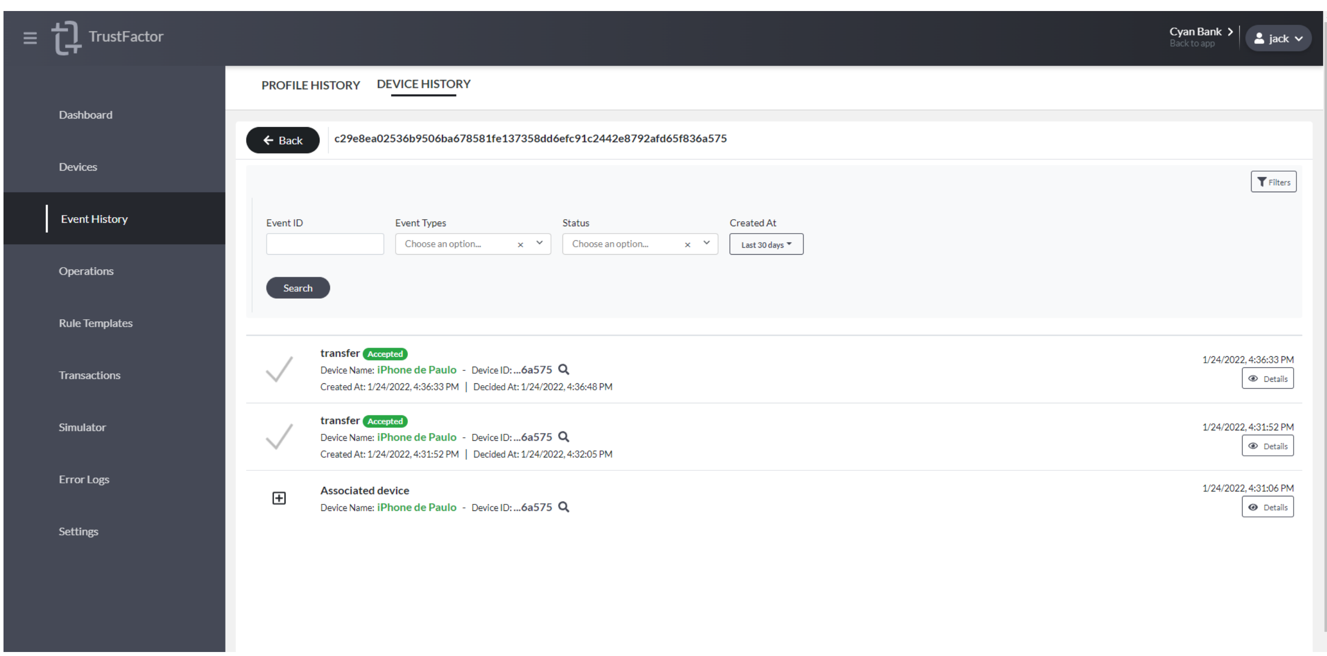
The device history is used to consult all the activity performed by a device.

Permissions required to view this screen: **- View Contracts - View Transactions**

Here you can search by device.



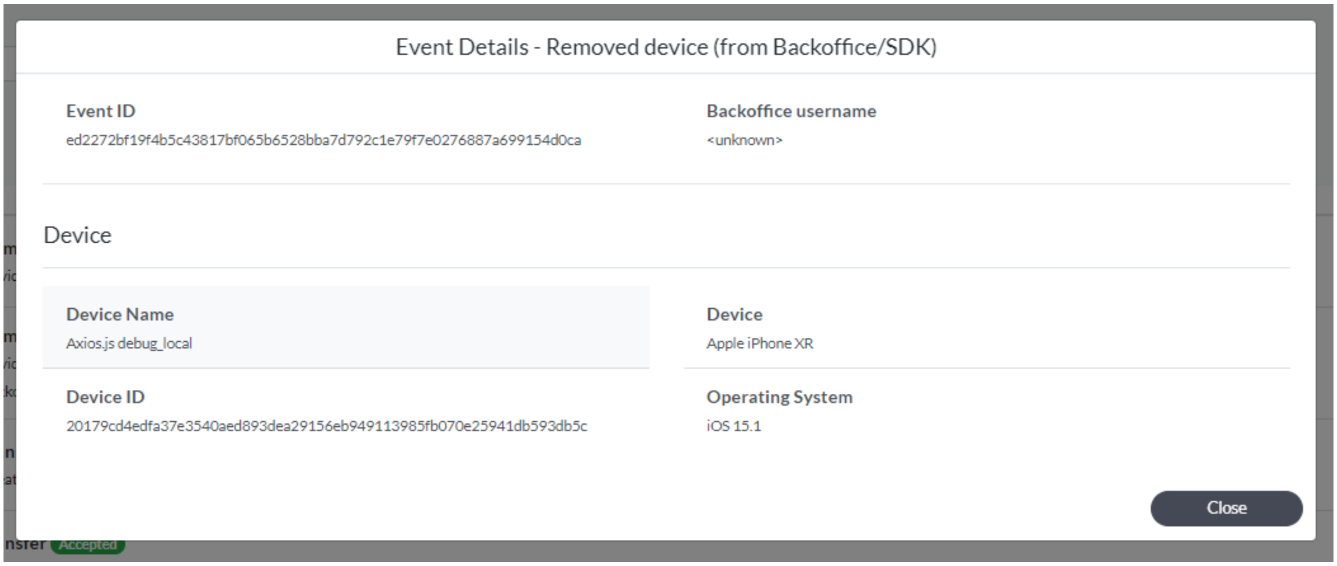
Here you can see device event list.



You can filter for:

- Event ID
- Event Types
- Status
- Created At (time period)

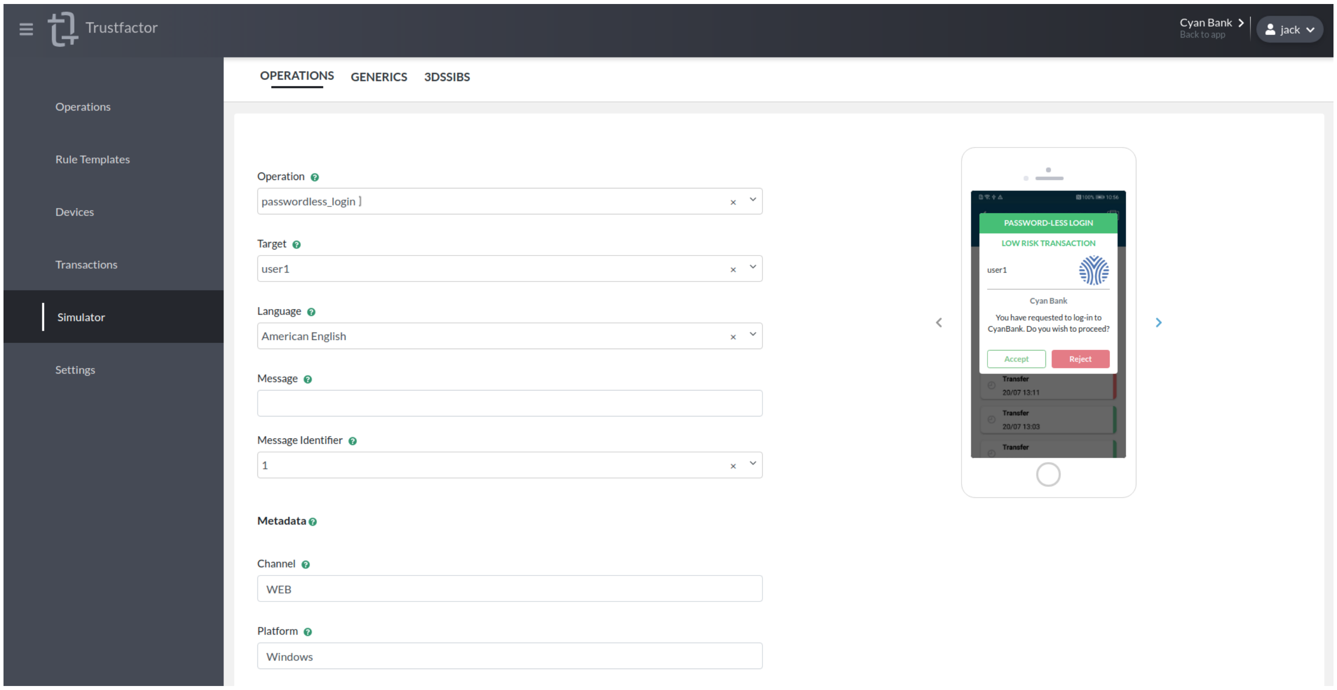
By pressing the “Details” button at the end of the table row, you will be able to see the event details.



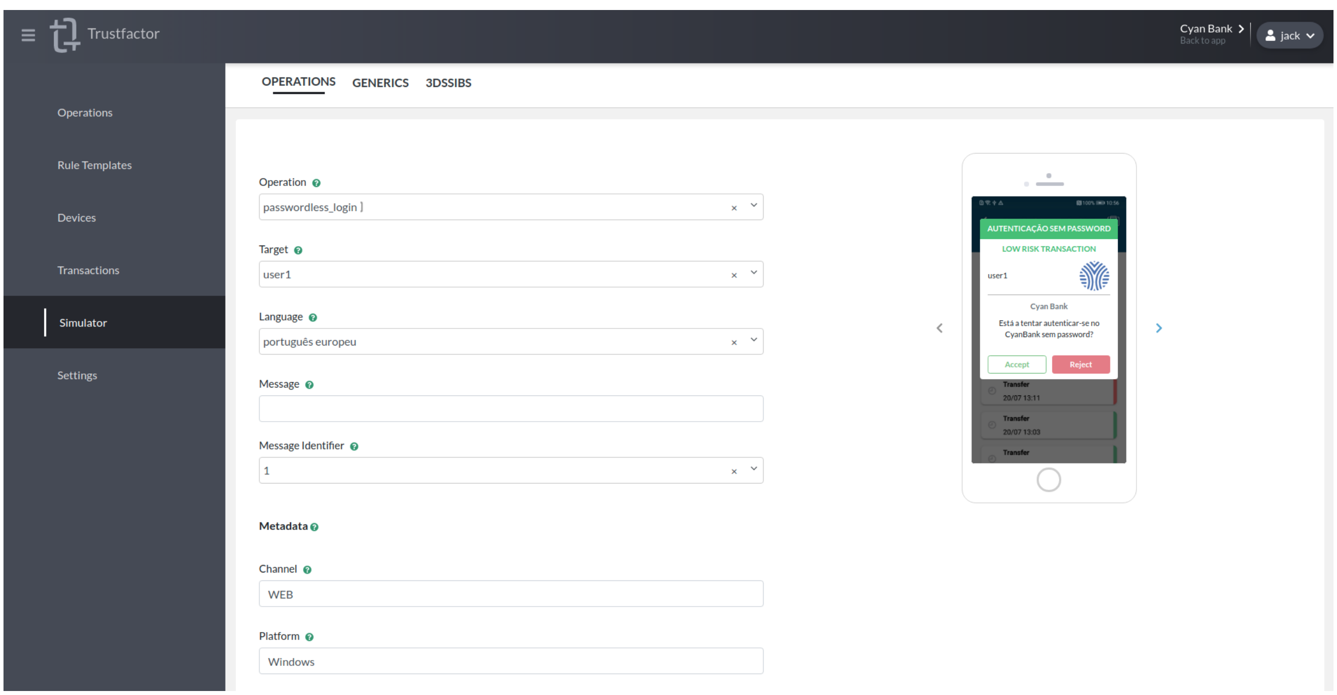
Simulator

In the simulator we can check what the user will see when they receive an authentication request for a given operation. The forms will match the API exposed through the SDK's CreateTransaction methods and will allow you to test what operations will look like before you start development.

Permissions required to view this screen: **- Manage Roles - View Contracts**



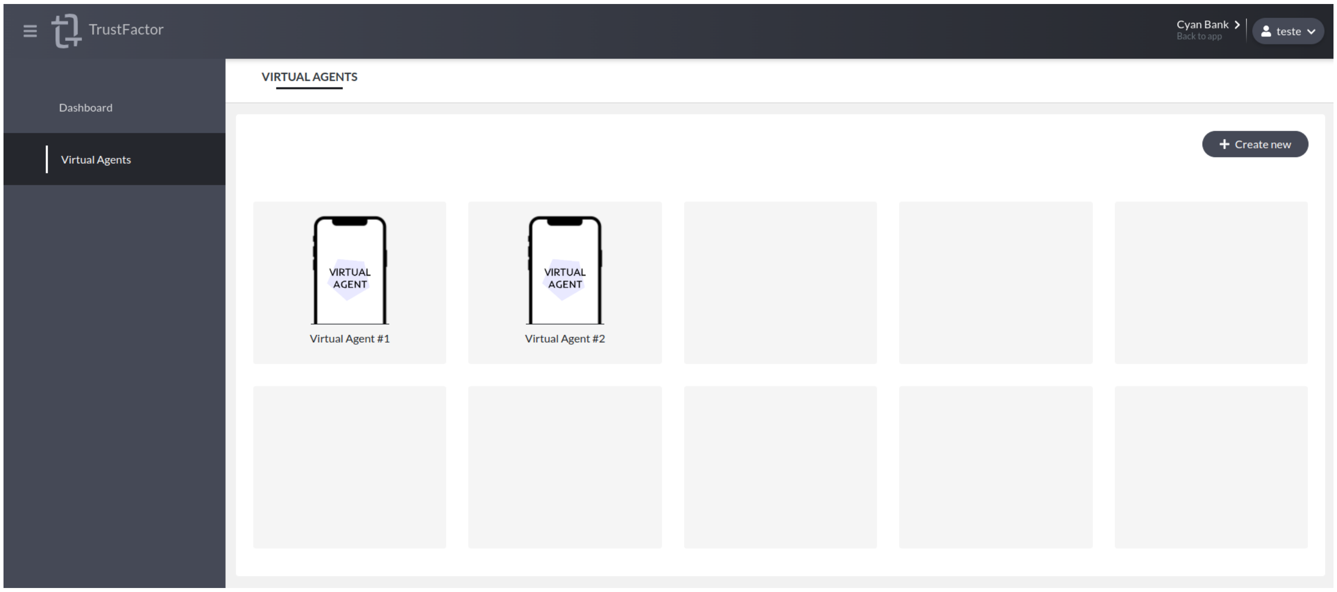
If we switch the device language to *Portuguese*, we will see it reflected in the simulator.



Virtual Agents

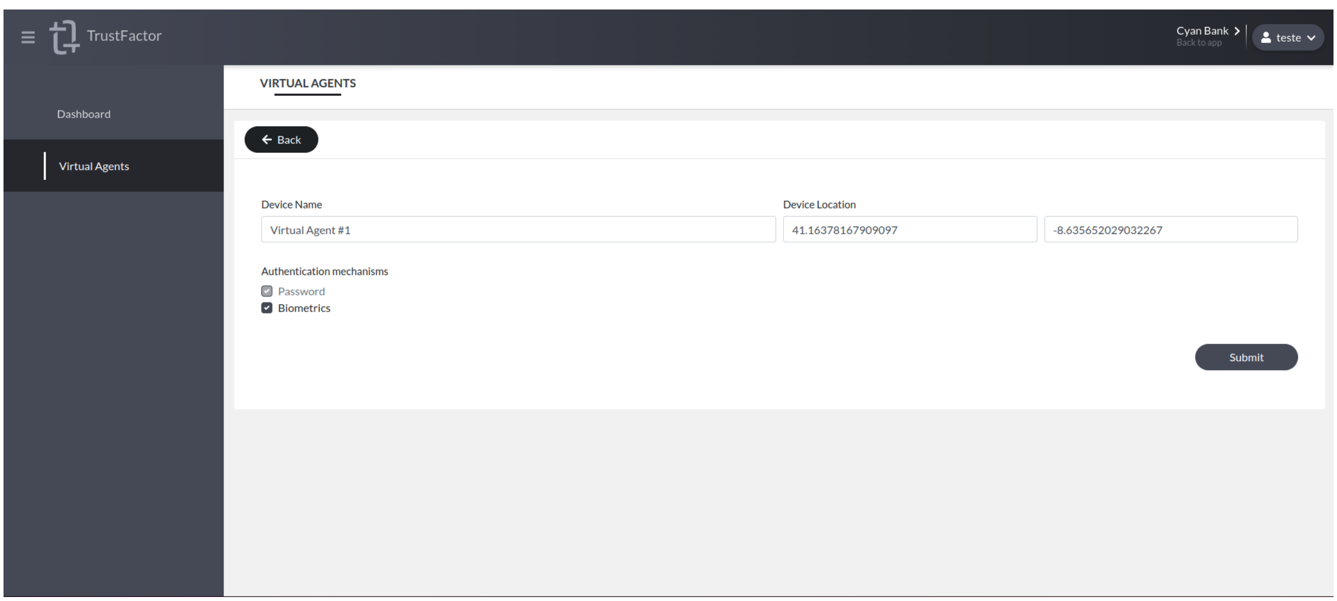
Virtual Agents main function is to emulate a real device, mainly for development use when a real mobile agent is not available. This functionality allows tests such as enrollment, transaction decision, profile sharing, backup and recovery.

Permissions required to view this screen: - **Use Virtual Agents**



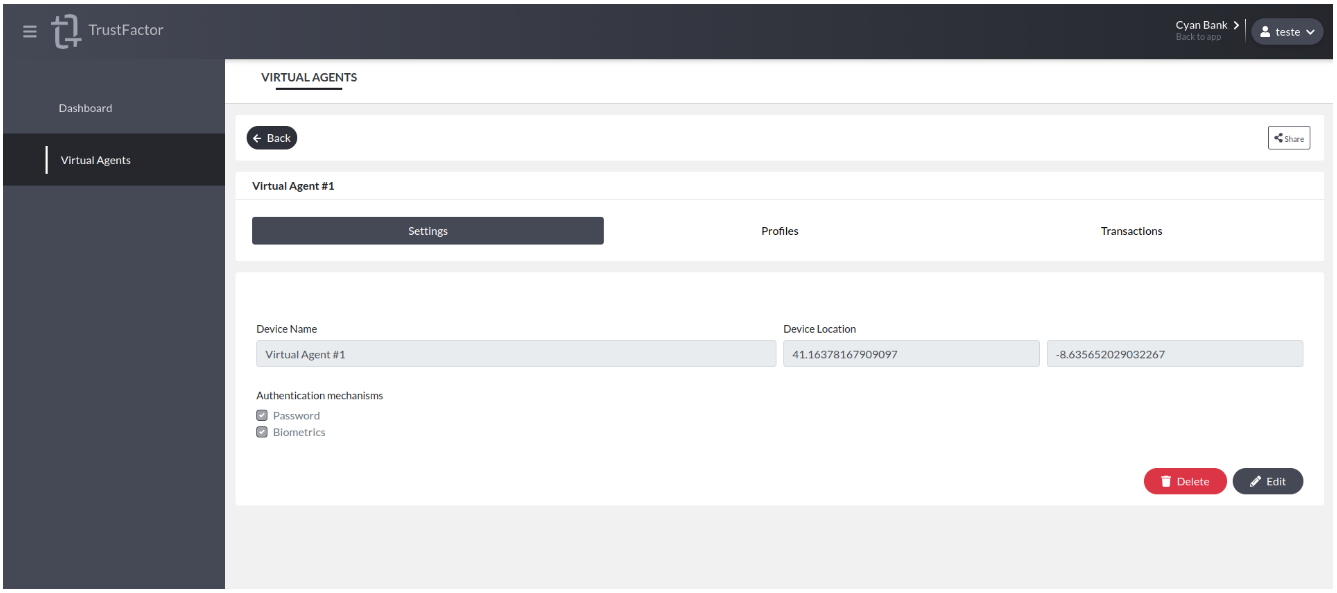
Create Virtual Agent

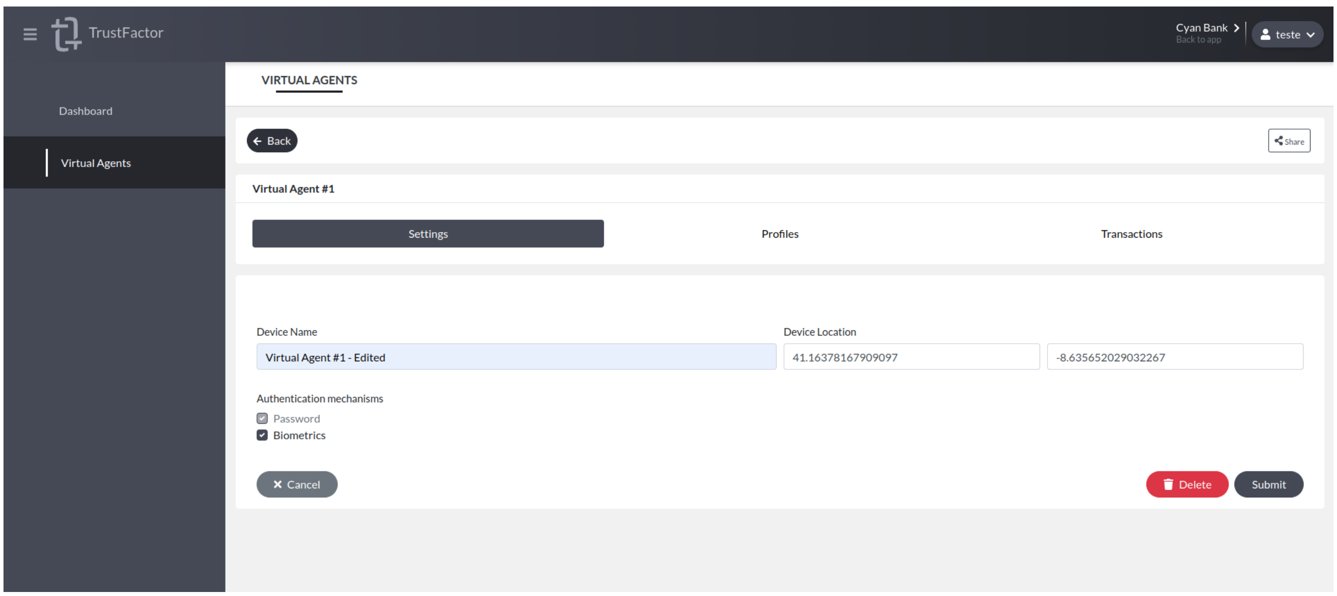
When creating a Virtual Agent, you are registering a "TFVA" device (that is the make and model shown in the *Devices* tab), and you can choose the location and authentication mechanisms (Biometrics and Password). This is akin to installing the mobile agent application on an Android or iOS device.



Virtual Agent Settings

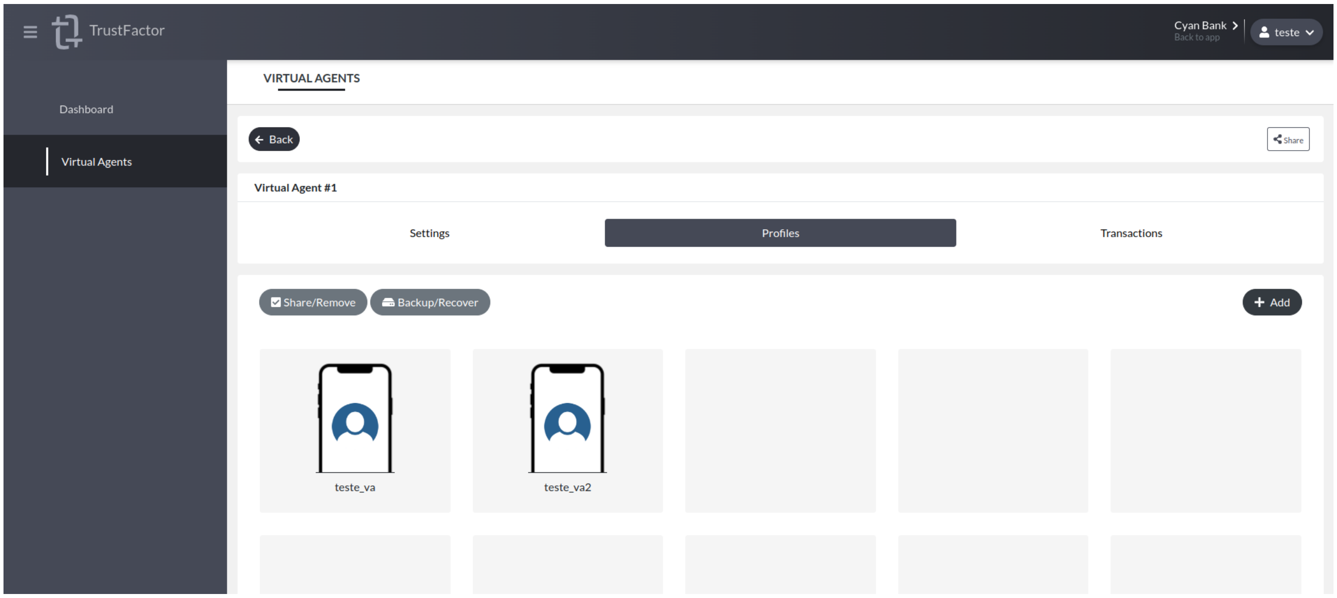
You will be able to view and edit the device settings if necessary.





Virtual Agent Profiles

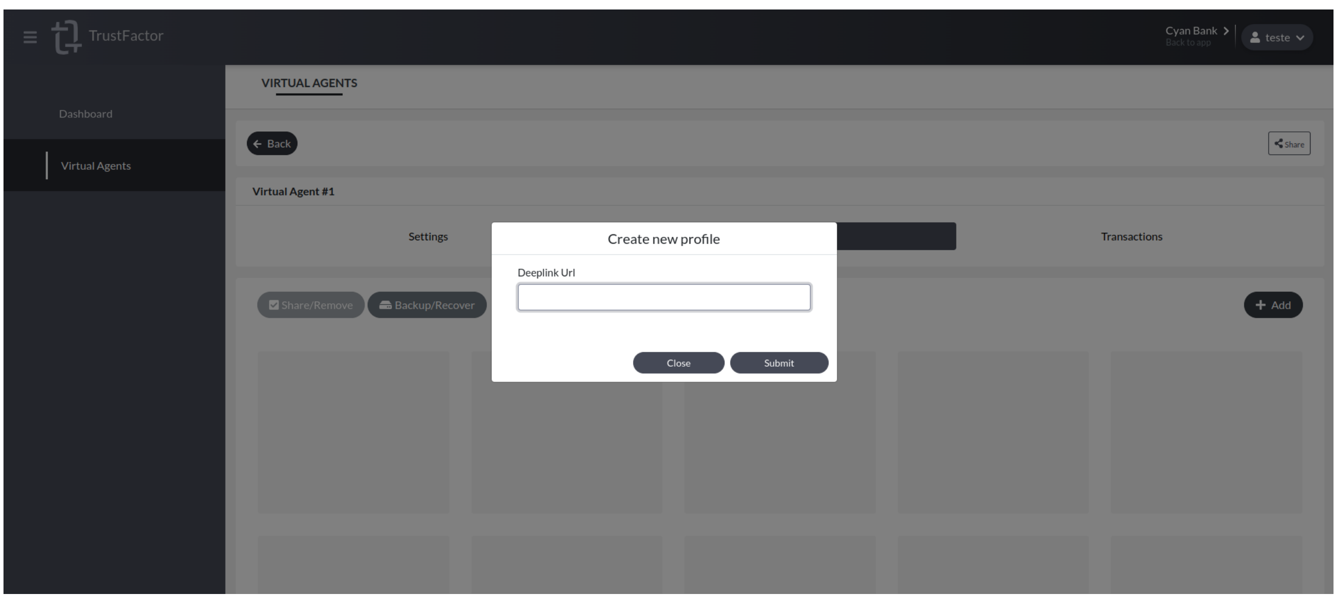
You will be able to view and manage profiles: register, share, remove, recover and create backup.



Virtual Agent Profiles Registration

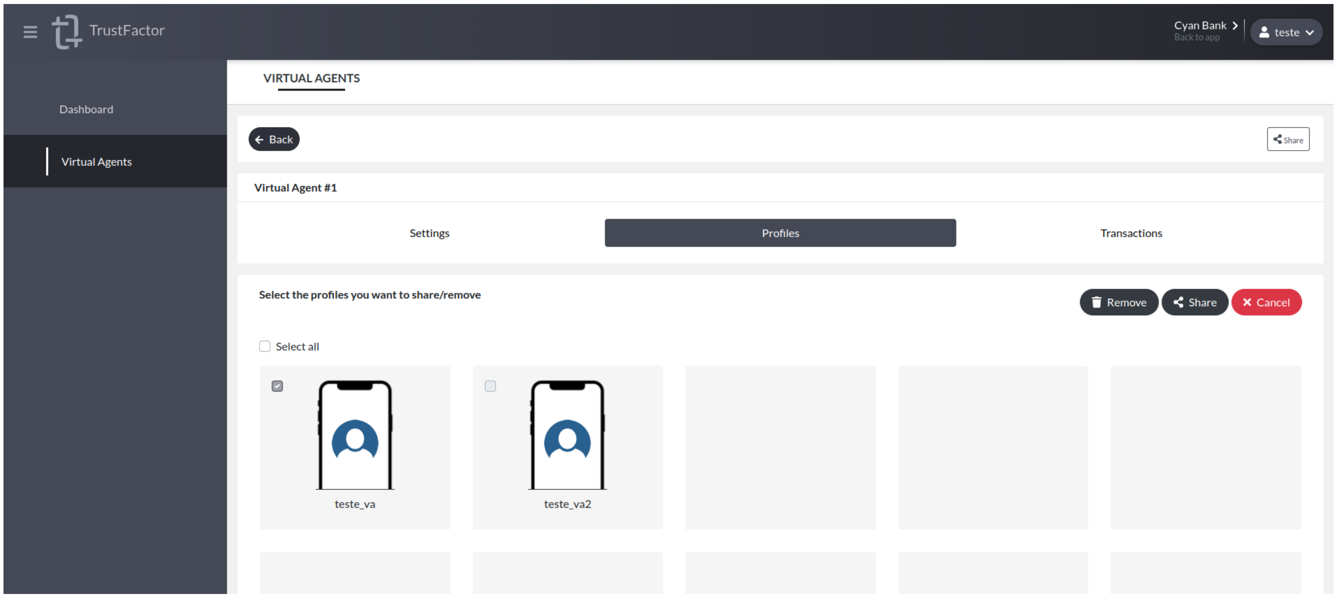
In order to enroll a virtual agent with a profile from your application, you must first use the SDK to obtain a registration deep link URL, just like in a real TrustFactor agent. This is akin to scanning a QR code or opening a registration deep link with the TrustFactor App.

To register the profile, open the Virtual Agent you wish to enroll and access the "Profiles" tab. Use the "Add +" button to open the input box where you can enter the deep link URL to complete the registration.



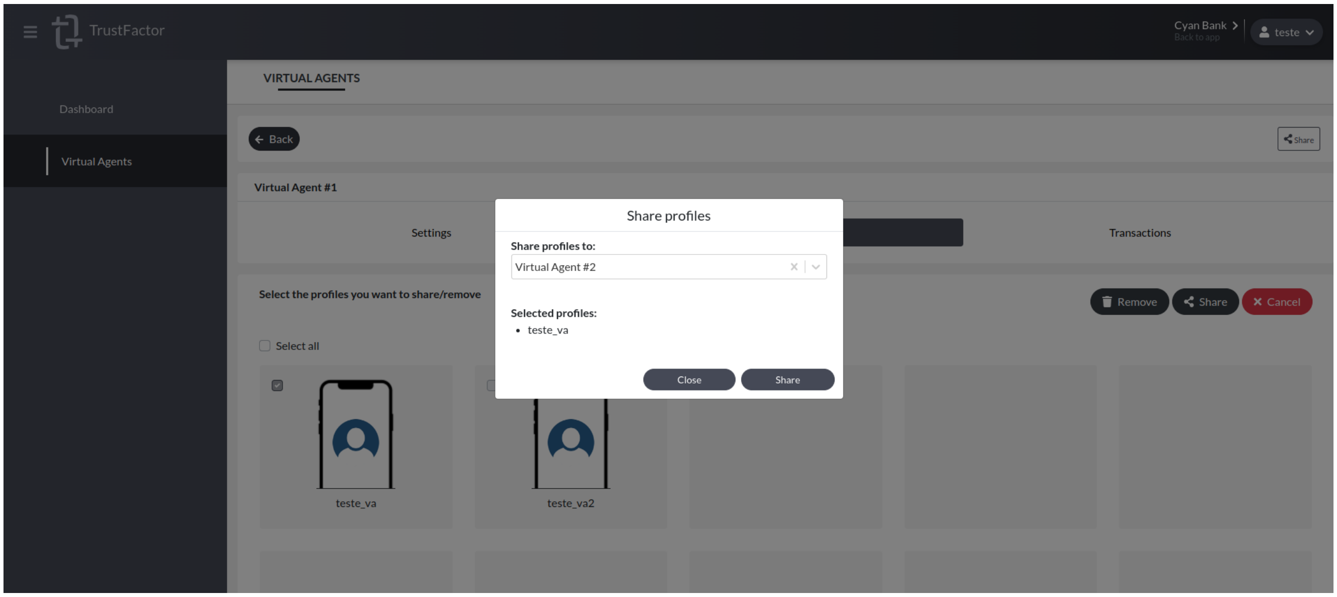
Virtual Agent Profiles Share/Remove

You can remove or share profiles between Virtual Agents, just like in the real agent apps.

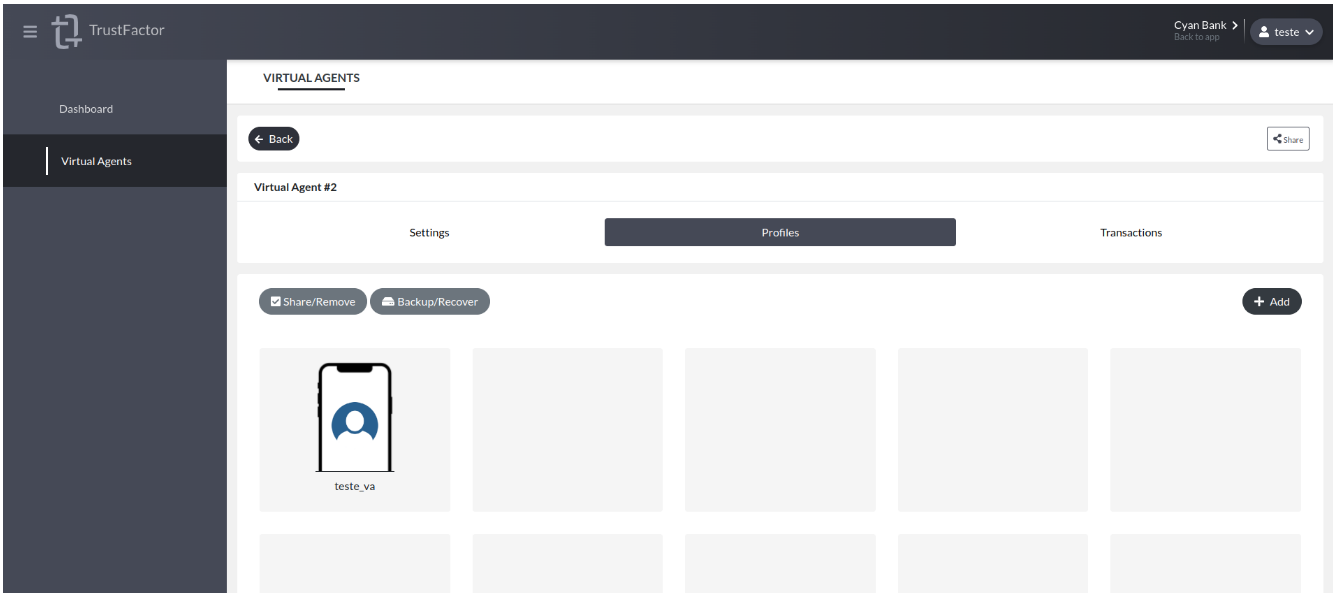


Remove To remove profiles from the Virtual Agent, click on the "Share/Remove" button, then select which profiles you want to remove and finally click on the "Remove" button.

Share To share profiles from the Virtual Agent, click on the "Share/Remove" button, then select which profiles you want to share, finally click on the "Share" button and select which Virtual Agent you want to share and click on "Share".

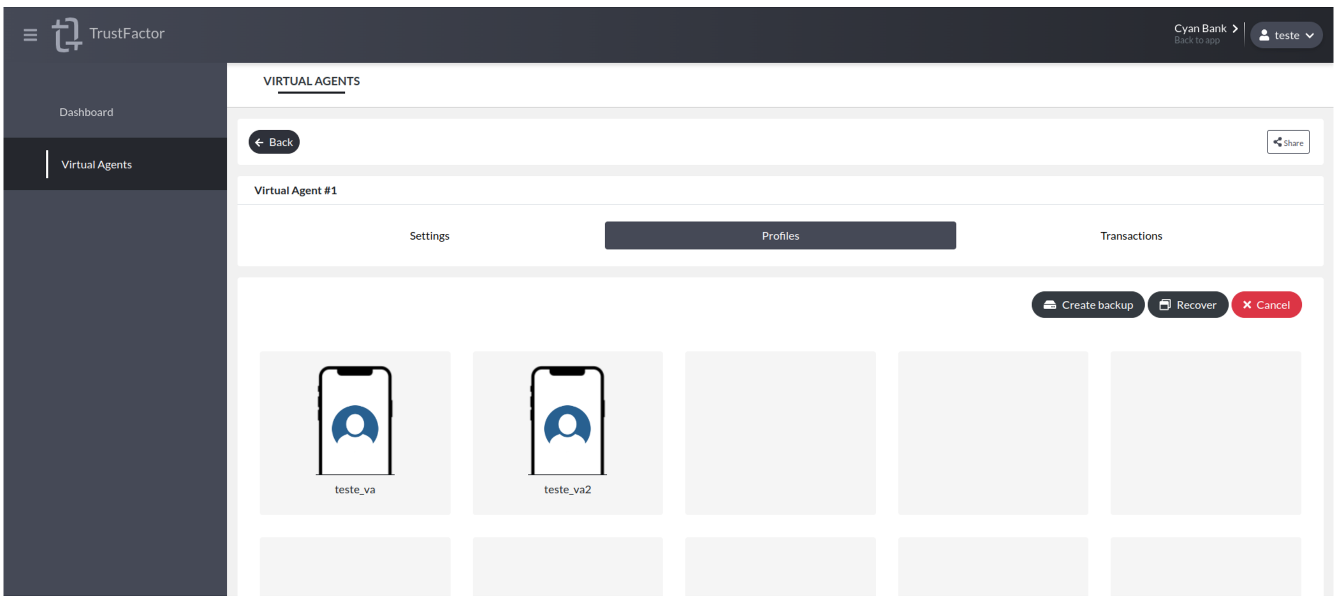


In the example you can see that in the Virtual Agent "Virtual Agent #2", it contains the profiles that were previously shared.

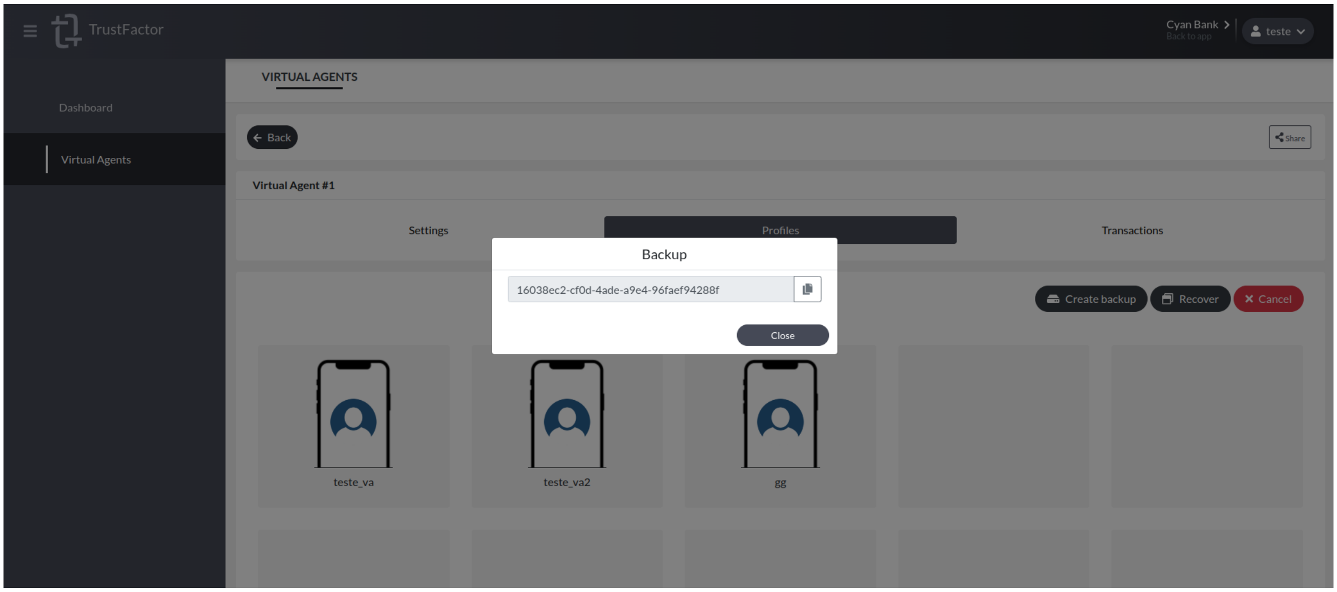


Virtual Agent Profiles Backup/Recover

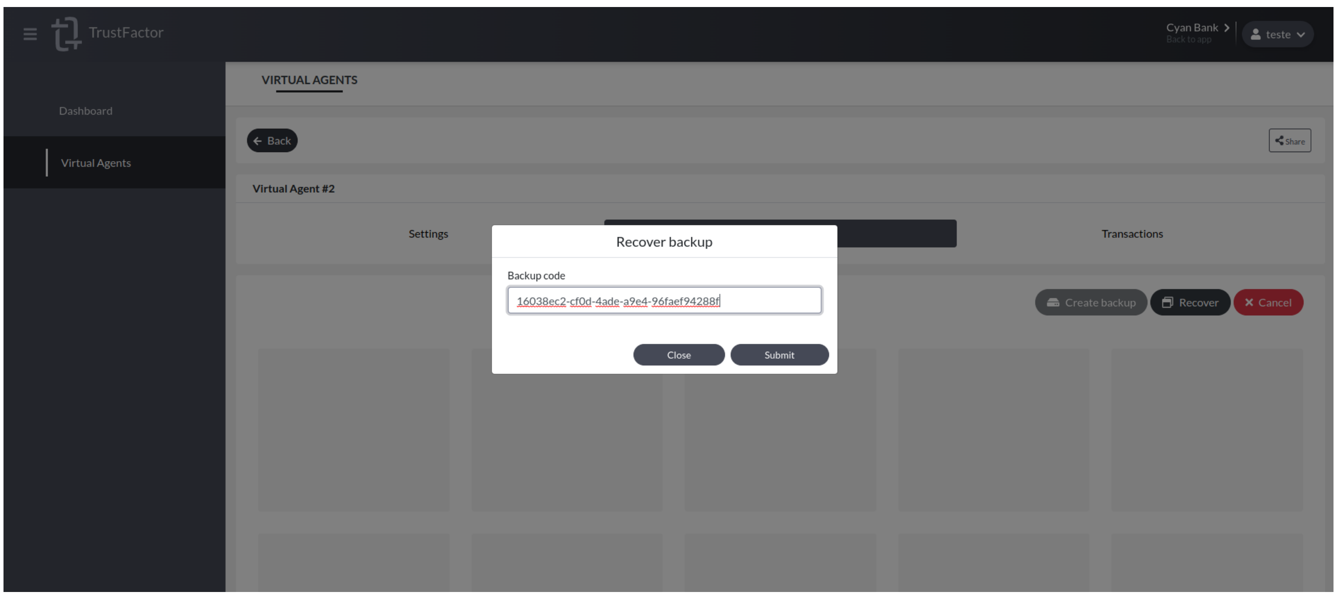
Here you can generate a backup or recover profiles (using the code generated by a backup).



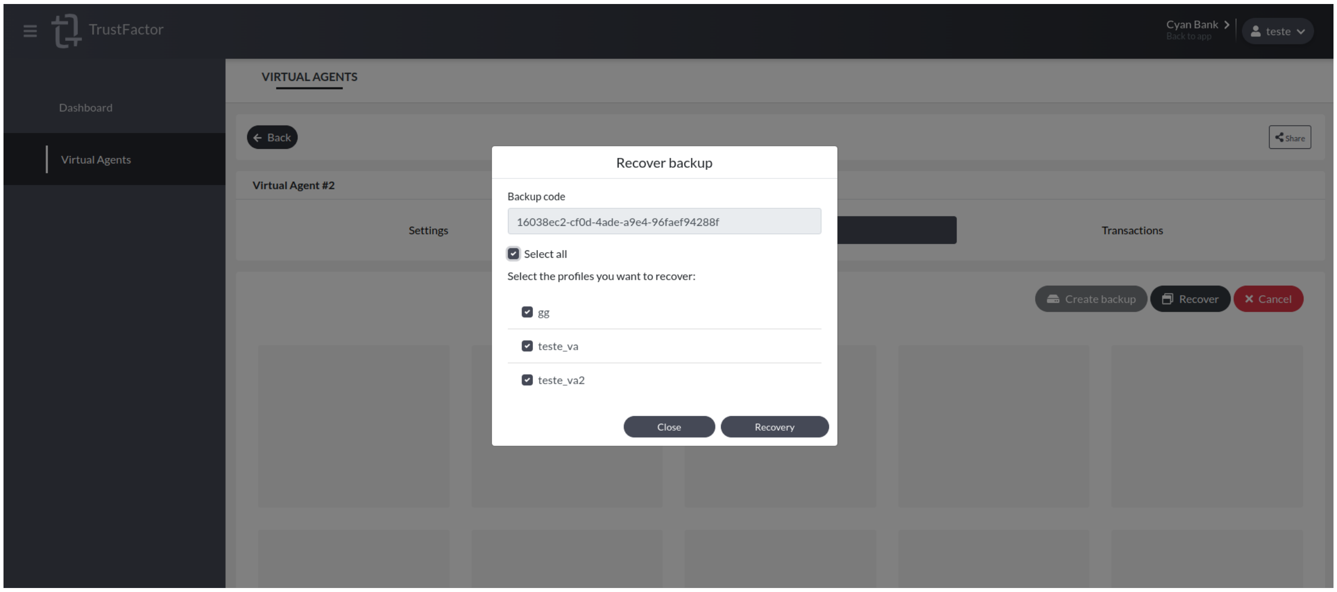
Backup To generate a backup, click on the “Backup/Recover” button and then on “Create backup” and wait until the request finishes and the recovery code is displayed.

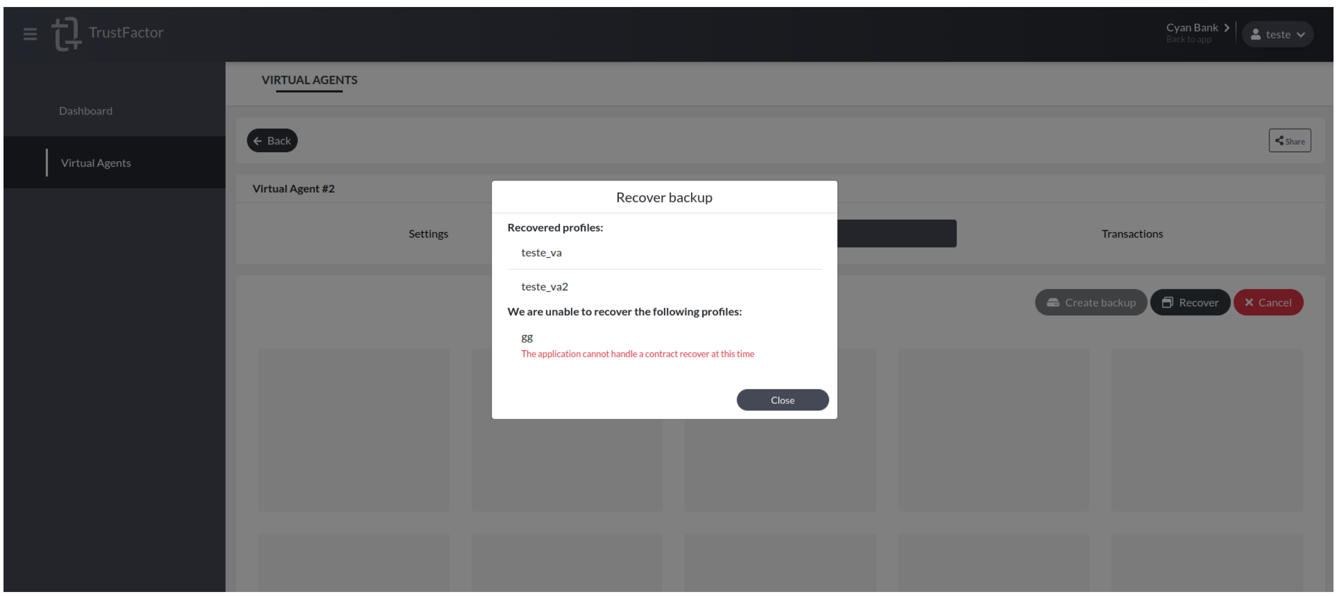


Recover To perform the recovery, press the “Backup/Recover” button, then “Recover” and enter the recovery code.



Then select the profiles you want to recover and wait for the response, if it is not possible to recover any profile, the reason will be displayed.


TrustFactor
Cyan Bank
Back to app
teste
VIRTUAL AGENTS
Dashboard
Virtual Agents
Back
Share
Virtual Agent #2
Settings
Transactions
Create backup
Recover
Cancel
Recover backup
Backup code
16038ec2-cf0d-4ade-a9e4-96faef94288f
Select all
Select the profiles you want to recover:
gg
teste_va
teste_va2
Close
Recovery


TrustFactor
Cyan Bank
Back to app
teste
VIRTUAL AGENTS
Dashboard
Virtual Agents
Back
Share
Virtual Agent #2
Settings
Transactions
Create backup
Recover
Cancel
Recover backup
Recovered profiles:
teste_va
teste_va2
We are unable to recover the following profiles:
gg
The application cannot handle a contract recover at this time
Close