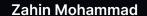


/activate/ Developer Conference 2022

Business Rules for Wallet Transactions

# Wallet Transactions: Policies and Pending Approvals



Senior Software Engineer BitGo



## Agenda

1/ Setup Prelude

2/ What is a Policy?

3/ Whitelist Policy Rule Anatomy

4/ Create Wallet & Transaction Exercise

5/ Create Policy Rules

Exercise

6/ Trigger Policy Rules

7 / Hacking Time / Q&A





@zahin\_mohammad

## Zahin Mohammad

Hey, I'm a software engineer based out of Toronto. Outside of BitGo, I like to experiment with DeFi protocols, travel and make coffee!

## **Setup Prelude**

Let's get Installing

Let's install some of the software in the background now before the intro.

Follow the Setup Guide Here:

https://github.com/zahin-mohammad/bitgo-activate-policies-workshop



## What is a Policy?

## What is a Policy?

#### A set of business rules on a wallet or enterprise

- Allows configurable rules that decide how to proceed with a transaction before it gets broadcasted
- It can be used as a tool to help prevent mistakes or bad actors

### Example

- "I want my team lead to approve all outgoing transactions"
- "I only want to allow withdrawals to known addresses"
- "Do not allow more than 10 BTC to be withdrawn from this wallet in a 24 hour period"

#### Reference

- https://developers.bitgo.com/api/v2.wallet.createpolicy
- https://developers.bitgo.com/api/v2.wallet.removepolicy
- https://developers.bitgo.com/api/v2.wallet.updatepolicy

## What Happens on a Policy Rule Violation?

#### It all depends on the action!

- Actions define what to do when a policy rule is violated
  - Ex. sending to to a non whitelisted address
- Options
  - 'deny', 'getApproval', 'getFinalApproval'
  - 'getApproval' or 'getFinalApproval' will create a pending approval
  - Deny will not create a pending approval and will deny the pending request from being processed

### **Pending Approval**

- An object responsible for tracking approvals on entity changes
  - Ex: adding users, adding policy rules, triggering policy actions
- Enables admins to manually verify policy violations or entity changes
  - An admin can either reject or approve a pending approval

#### Reference

- https://developers.bitgo.com/api/v2.approval.get
- https://developers.bitgo.com/api/v2.approval.update

## Policy Rules

## **Anatomy of a Policy**

```
date: '2022-06-16T17:49:34.803Z',
id: '62ab6d2e7eeda000072123101257347a',
label: 'default',
rules: [Array],
version: 7,
latest: true
```

## **Anatomy of a Policy Rule**

```
id: 'my activate whitelist policy rule',
lockDate: '2022-10-21T16:19:07.226Z',
type: 'advancedWhitelist',
action: { type: 'deny', userIds: [] },
condition: { entries: [Array] }
```

## Whitelist Policy Rule Anatomy (Condition)

```
add: {
    item: "2mL1GH2KAevXzCQ7qVZGeRUd1C1uzrmNvZ5AysK1WPJm",
    type: "address",
```

#### Reference

- https://developers.bitgo.com/api/v2.wallet.createpolicy
- https://developers.bitgo.com/api/v2.wallet.removepolicy
- https://developers.bitgo.com/api/v2.wallet.updatepolicy



### **Exercise time! Create a Wallet**

#### yarn run manage-wallet

- Create a file called `scratchpad.txt`
  - Note down your enterprise ID
    - You can log into the webapp at app.bitgo-test.com to copy it
  - Email and Password of user 1
  - Email and Password of user 2
  - As the demo continues, any other useful ld's (wallets, pending approvals)
- Create a wallet using the `Create a New Wallet` command
  - Once a wallet is created, fund the "receive address" of the wallet with 1 SOL
- In a separate terminal window go through the `Accept a Wallet Invite` flow as user #2

## **Exercise time! Create a Transaction**

#### yarn run manage-wallet

- Login in as user 1
- Select option "Send a Transaction"
- Copy the walletId into the prompt
- What happened?
- Note: "2mL1GH2KAevXzCQ7qVZGeRUd1C1uzrmNvZ5AysK1WPJm" as the destination (or another wallet)

## **Exercise time! Create a Whitelist Policy Rule**

#### yarn run policy-exercise

- Implement the function startWhitelistPolicyExercise
  - View the code comments for hints and details
- Save and re-run
  - Select option "Create Whitelist Rule"
  - If all works out well, you should have a pending approval ID in your console! Note it down in your scratchpad
- Implement the function approvePendingApproval
  - View the code comments for hints and details.
- Save and re-run in a new terminal window!!
  - Login as a different user!!
  - Select option "Approve Pending Approval"
- Ask a TA if you are running into any issues
  - Or view the "solution" branch to see the completed solution
  - git checkout solution

## **Exercise time! Whitelist Violation**

#### yarn run manage-wallet

- Select option "Send a Transaction"
- Send to an address outside of what you whitelisted previously!
  - You can use
     "GhrweBg5xzESfNv1v6fTtY7Lht6puoXZJEXjzL7AQnbP"
  - What happened?
- Repeat the above but send to you whitelisted address

## **Next Steps**

- There are a few other tasks in `manage-policy.ts`
  - Try it yourself!
- With your forked repository, you can use this as a base to prototype your new product built with BitGo
- Refer back to <a href="https://developers.bitgo.com/">https://developers.bitgo.com/</a> if you ever get stuck
  - Or ask a TA

## Questions?

@ActivateDev / #ActivateDevCon

