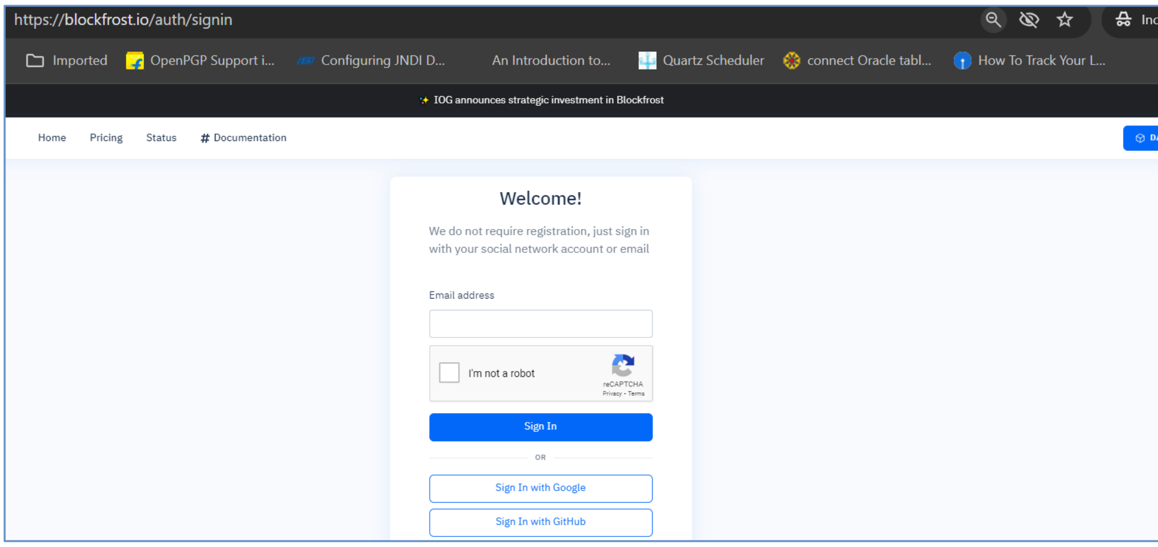
Minswap Cardano Connector usage Guide:

Minswap Cardano connector uses the following libraries/utilities to connect to Cardano and perform the operations.

1. Blockfrost API – It is an API as a service to interact with Cardano blockchain and its ecosystems. In order to use the API user need to create an account and configure a project (create a project_id) as below:
   a. Open the url(**https://blockfrost.io/auth/signin**) and login using your credentials(if you donot have the login you first need to create an account from this link).



https://blockfrost.io/auth/signin

Imported | OpenPGP Support i... | Configuring JNDI D... | An Introduction to... | Quartz Scheduler | connect Oracle tabl... | How To Track Your L...

IOG announces strategic investment in Blockfrost

Home | Pricing | Status | # Documentation

Welcome!

We do not require registration, just sign in with your social network account or email

Email address

I'm not a robot

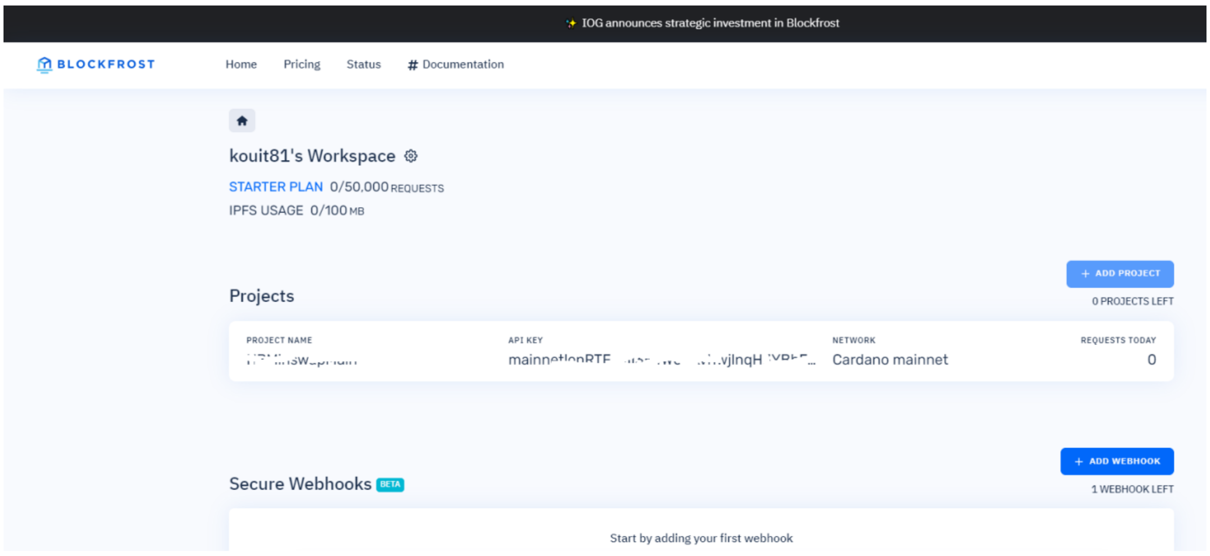
Sign In

OR

Sign In with Google

Sign In with GitHub

   b. After the login go to the Dashboard(https://blockfrost.io/dashboard). Option present at the top right corner just after the login.The page will look like below:



IOG announces strategic investment in Blockfrost

BLOCKFROST | Home | Pricing | Status | # Documentation

kouit81's Workspace

STARTER PLAN 0/50,000 REQUESTS

IPFS USAGE 0/100 MB

Projects

+ ADD PROJECT

0 PROJECTS LEFT

PROJECT NAME	API KEY	NETWORK	REQUESTS TODAY
[redacted]	[redacted]	Cardano mainnet	0

Secure Webhooks BETA

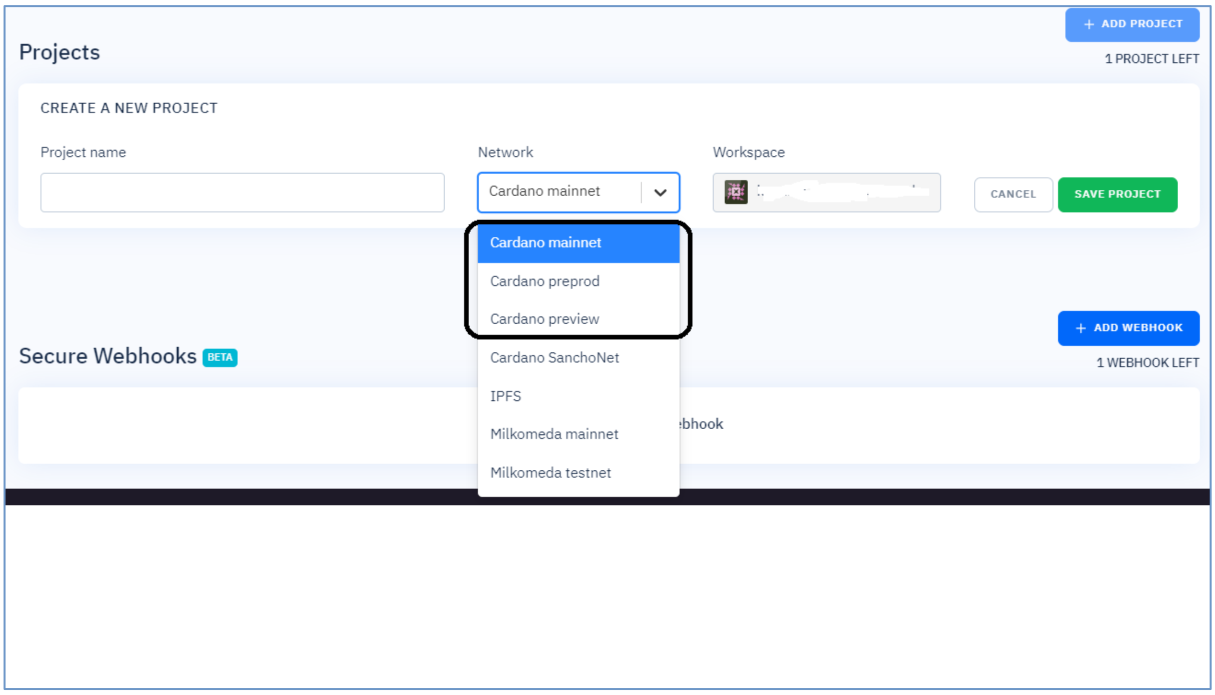
+ ADD WEBHOOK

1 WEBHOOK LEFT

Start by adding your first webhook

c. In the dashboard there are option to create a new project and also to register for any webhook in case you want to get notification for any event. For our usage we need to create a project by clicking +**Add Project** button

d. While creating the project please ensure to select the appropriate Network to connect via this project as shown below:



Projects

1 PROJECT LEFT

CREATE A NEW PROJECT

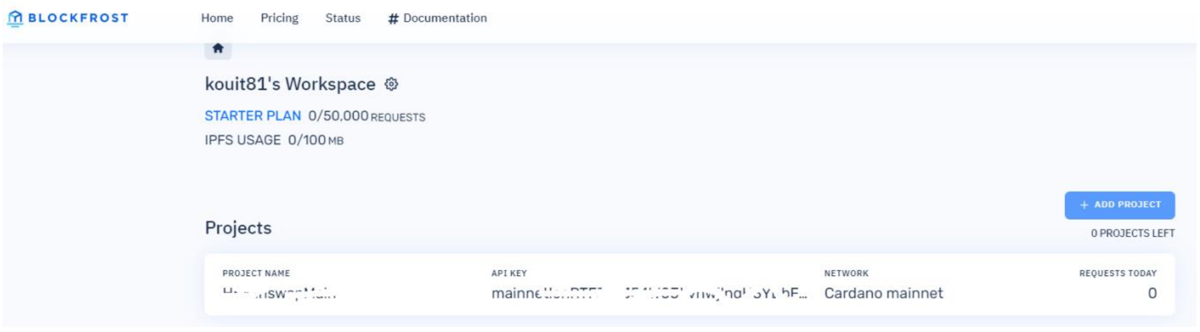
Project name	Network	Workspace
	Cardano mainnet	[redacted]

Secure Webhooks BETA

1 WEBHOOK LEFT

bhook

e. After the project is created it will look like below, please copy the API KEY(also called project_id) which we would need to use in our connector.



kouit81's Workspace

STARTER PLAN 0/50,000 REQUESTS

IPFS USAGE 0/100 MB

Projects

0 PROJECTS LEFT

PROJECT NAME	API KEY	NETWORK	REQUESTS TODAY
[redacted]	[redacted]	Cardano mainnet	0

2. Minswap SDK & lucid-cardano library

Minswap SDK github link is : https://github.com/minswap/sdk, however as this package uses lucid-cardano as dependency and which is an ESM package, so it's also an ESM package.However as Hummingbot Gateway is CJS package compatible, hence in order to make the project build compatible to CJS the lucid-cardano and minswap/sdk libraries were converted to CJS package.In order to be able to use the Minswap cardano Connector in Hummingbot Gateway you need to follow certain steps to configure the application before using the API calls.

yarn copyLib – it copies the required libraries from lib folder

Note Babel is used to convert the compatible CJS package.

The converted lucid-cardano CJS package is available in lib folder. You need to have yarn dependency installed to build Hummingbot project. So please execute the **yarn copyLib** command before building the Hummingbot gateway project.

3. Installing Hummingbot Client and Hummingbot Gateway

In order to use the Hummingbot you need to first install it from its website following the instruction,[use the install from source option]

The Client can be installed as specified in this link:

- **Source**
  - https://hummingbot.org/installation/

In order to connect to different DEX/Connector Hummingbot Gateway also need to be installed

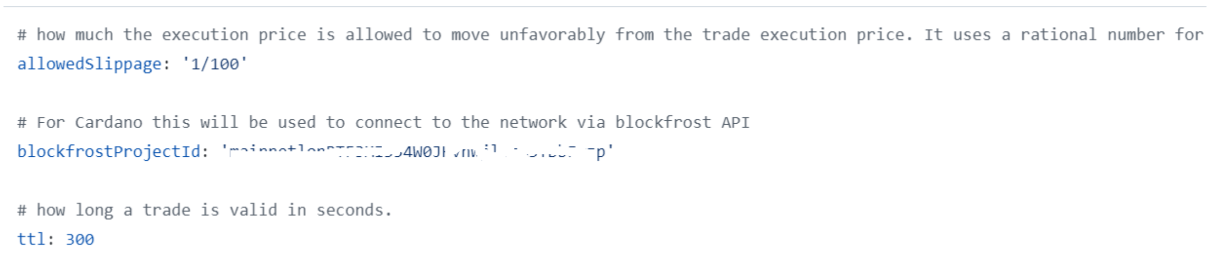
- ❖ **Gateway** source code link with minswap implementation:
  https://github.com/AIQUANT-Tech/HummingbotGateway/tree/main

  The installation process is almost similar to the one mentioned in the link :
  (https://hummingbot.org/gateway/installation/#install-from-source)
  The only difference being just before running **yarn build** command please execute **yarn copyLib** command to copy the library.

  So the command execution order would be:
  git clone > yarn > yarn copyLib >yarn build

  and then follow the instruction as it is in the official website link.

- ❖ Before Gateway module can be started using the steps mentioned in the link(https://hummingbot.org/gateway/installation/#start-instance), please make sure to add the blockfrost API Key in your cardano configuration file inside the project location : *src/templates/cardano.yml.*
- ❖ The Gateway project source can be downloaded from the following location which is a fork from the original git repo and then Minswap related part added to it.
  https://github.com/AIQUANT-Tech/HummingbotGateway/tree/main



```
# how much the execution price is allowed to move unfavorably from the trade execution price. It uses a rational number for
allowedSlippage: '1/100'

# For Cardano this will be used to connect to the network via blockfrost API
blockfrostProjectId: '[redacted]p'

# how long a trade is valid in seconds.
ttl: 300
```

You need to edit this file and replace the value of **blockfrostProjectId** with your API Key, you can also change the values of **allowedSlippage** and **ttl** values as per your strategy.

4. Creating an Wallet

- ✓ In order to be able to use the Hummingbot Gateway to query/transaction through Minswap Connector user need to have an **wallet**(like Yoroi etc.) with a **receiver address** and the wallet **seedphrase** handy, seedphrase would be used to place a trade in Minswap; For example if the user already doesnot have an wallet already, browser wallet can be installed and get started easily.

5. Connecting Minswap DEX and querying/transacting via the Hummingbot Gateway API

a. Currently Minswap Connector supports the following endpoints:

i. **/amm/price** –

this API endpoint is used **to get the Buy/Sell price of MIN-ADA pair**.
The input parameters are ,
quote=<token_symbol,**MIN**>
base=<token_symbol,**ADA**>
amount=<request_amount,number,100>
side=BUY/SELL
chain=***cardano[this hardcoded value need to be passed as chain]***
network=mainnet/preprod
connector=***minswap[this hardcoded value need to be passed as connector]***

poolId=<minswap pool id**>** <!—this is an optional input →

The logic and expected behavior is outlined below:

when, side = BUY

Expected output: **To BUY** 100 ADA, calculate amount of MIN required

when, side = SELL

Expected output: **When I SELL** 5 ADA how many MIN can I get in return

In the response, the values will come in the below output parameters,

expectedAmount : This is the total converted amount as output
price : This is conversion factor

ii. **/amm/trade** –

this API endpoint is used to,

- **Place** a BUY/SELL **trade** in Minswap
- **Cancel** a BUY/SELL **trade** order in Minswap given the transaction hash of the original order

When side BUY,
it means : acquire particular requested amount(amount) of base(ADA) by selling quote(MIN)
When side SELL,
it means: sell particular requested amount(amount) of base(ADA) to acquire quote(MIN)

For Cancelling the transaction, the following optional parameters were introduced in TradeRequest interface under amm.requests on top of what the service endpoint already supported for other chains,

isCancelled? : boolean;//to identify if to cancel txn for cardano
txnHash?: string; //this is the transaction identifier cancel txn for Cardano

The input parameters are ,
quote=<token_symbol,**MIN**>
base=<token_symbol,**ADA**>
amount=<request_amount,number,100>
side=BUY/SELL
chain=***cardano[this hardcoded value need to be passed as chain]***
network=mainnet/preprod

connector=***minswap[this hardcoded value need to be passed as connector]***
poolId=<minswap pool id**>**
address=<this is the wallet address**>**
allowedSlippage=<this is optional, if not present the configured value will be used**>**
seedPhrase=<seed phrase to connect wallet to the chain/network>
isCancelled=<Boolean,to indicate if the request is to Cancel a trade>
txnHash=<transaction hash for cancelling trade order>

The logic and expected behavior is outlined below:

<u>for BUY/SELL trade</u>

Code/Logical flow:

a. Create a blockfrost adapter instance using the API Key configured
b. Create a lucid instance and connect to Cardano network using Lucid and blockfrost library
c. Use the seedphrase/mnemonic to connect the wallet to the Lucid instance
d. Create an instance of MinSwap using the configured Lucid instance
e. Calculate the UTXOS at the specified address
f. Specify the input amount and asset (e.g., ADA) and the output asset to swap for
g. Call the following minswap libraries based on BUY/SELL,
    i. **BUY** - execute *<u>calculateSwapExactOut</u>*(Calculate necessary Amount In & Price Impact to cover the @exactAmountOut while swapping exact out)
    ii. **SELL** – execute *<u>calculateSwapExactIn</u>*(Calculate Amount Out & Price Impact while swapping exact in)
h. Create Swap Transaction
i. Complete the transaction using Lucid Cardano instance.
j. Sign the transaction with the user's wallet.
k. Submit the signed transaction to the Cardano network.

In the response, **txnHash** will be returned for a successful transaction which can be used for any further reference.

<u>In case of a Cancel order,</u>

a. **isCancelled** and **txnHash** input are mandatory.

b. For Cancelled order this API will use **buildCancelOrder** method inside minswap SDK library to build a Cancel transaction using the transaction hash of the original order passed in the request.
c. Complete the transaction using Lucid Cardano instance.
d. Sign the transaction with the user's wallet.
e. Submit the signed transaction to the Cardano network.

iii. **/amm/liquidity/add**-

This endpoint is used to add liquidity in the specific MIN/ADA liquidity pool.Minswap SDK is used to create and submit a withdrawal transaction from a liquidity pool on the MinSwap decentralized exchange (DEX) with the following steps:

Code/Logical flow:

a. Create a blockfrost adapter instance using the API Key configured
b. Create a lucid instance and connect to Cardano network using Lucid and blockfrost library
c. Use the seedphrase/mnemonic to connect the wallet to the Lucid instance
d. Create an instance of MinSwap using the configured Lucid instance
e. Calculate the UTXOS at the specified address
f. Use the **calculateDeposit** method in minswap sdk to calculate LP Amount while depositing,it takes as input the depositedAmountA and depositedAmountB and corresponding pool reserves and available liquidity and this returns the amount needed of Asset A and Asset and LP Token Amount you will receive.
g. It then creates a deposit transaction using Lucid and minswap sdk.
h. Complete the transaction using Lucid Cardano instance.
i. Sign the transaction with the user's wallet.
j. Submit the signed transaction to the Cardano network.

iv. **/amm/liquidity/remove** –

This endpoint is used to take out(remove) liquidity from the specific MIN/ADA liquidity pool. To pass the withdrawal amount the input parameter **decreasePercent** has been utlised for Minswap connector.It allows user to withdraw specific amounts of tokens from a liquidity pool, allowing users to reclaim their liquidity and any associated earnings.

To facilitate this transaction, **calculateWithdraw** method from minswap SDK has been used to _*calculate amount A and amount B after withdrawing @withdrawalLPAmount out of Liquidity Pool.It returns the amount A and amount B user will receive given the current liquidity of the specified pool and the waithdrawal amount passed as input.*_