Instruction

Transfer dfinity → polygon

1. Call the signed "approve" method in the dfinity token smart contract (https://ic.rocks/principal/oh7zz-gyaaa-aaaai-qfm3a-cai)

2. Send a post request to **/api/v1/save-transaction**

Request example:

```
{
 "sender": string,//sender wallet address
 "senderType": string,//sender type. Value “dfinity” or ”polygon”
 "amount": string,//shipping amount. Multiplied by 10^8. Example. If you
                          transfer 1 WICP, then you need to send 100000000
 "recipient": string,//recipient wallet address
 "recipientType": string /// recipient type. Value “dfinity” or ”polygon”
}
```

Transfer polygon → dfinity

1. Call the signed "**approve**" method in the polygon token smart contract (https://mumbai.polygonscan.com/address/0x8BC753423c58feEacE949000c9Cf4A9C9CD4909A)

2. Call the signed "**requestBridgingToStart**" method in the polygon bridge smart contract (https://mumbai.polygonscan.com/address/0x6e9b5E5D44D09f0c86aF2e6fED622F97246C700a). Save the hash of the transaction. Required when sending a request to the backend

3. Send a post request to **/api/v1/save-transaction**

Request example:

```
{
 "sender": string,//sender wallet address
 "senderType": string,//sender type. Value “dfinity” or ”polygon”
 "amount": string,//shipping amount. Multiplied by 10^8. Example. If you
                          transfer 1 WICP, then you need to send 100000000
 "recipient": string,//recipient wallet address
 "recipientType": string /// recipient type. Value “dfinity” or ”polygon”
  "polygonTransactionId": string // the hash of the transaction after
calling Bridging To Start method on the bridge polygon.
}
```

Getting a list of transactions by wallet address

Send a GET request to **/api/v1/%addressWallet%/transactions**

%addressWallet% - address wallet

Backend structure

Controller - accepts and processes post/get requests that come from the front. Calls a method on the Service

Service - main logic of the bridge. Here, transactions are recorded in the database, and the necessary methods are called for contracts.

PolygonListener - listens for polygon transactions. If a transaction is successful, calls the methods of the dfinity contract

CheckTransactionService — checking polygon transactions on a schedule. Checks polygon transactions if the PolygonListener fails

PolygonConnector - the logic of working with polygon contracts

DfinityConnector - the logic of working with dfinity contracts