**MyGatekeeper.js**

Location: **/home/demoadmin/Migration\_from\_Eris/gatekeeper**

Port: **3002**

Post Request Endpoint: **/MyGatekeeper**

FormData: Input Should be a JSON object. Note the difference between the gatekeeper javascript and this javascript are the fields “MyGatekeeperAddr” and “MyValidatorList”. *In particular, note that the field isHuman is not needed. This is hardcoded to be false.*

|  |  |
| --- | --- |
| **Field** | **Description** |
| “MyGatekeeperAddr” | A string of the user’s MyGatekeeper contract address |
| “pubKey” | String |
| “sig” | String |
| “uniqueId” | String |
| “uniqueIdAttributes” | String (delimiter “,”) |
| “ownershipId” | String |
| “ownershipIdList” | String (delimiter “,”) |
| “controlId” | String |
| “controlIdList” | String (delimiter “,”) |
| “ownershipTokenId” | String |
| “ownershipTokenAttributes” | String (delimiter “,”) |
| “ownershipTokenQuantity” | String (delimiter “,”) |
| “controlTokenId” | String |
| “controlTokenAttributes” | String (delimiter “,”) |
| “controlTokenQuantity” | String (delimiter “,”) |
| “identityRecoveryIdList” | String (delimiter “,”) |
| “recoveryCondition” | Int |
| “yesVotesRequiredToPass” | Int (recommended to be 2) |
| “validatorList” | String of public key hashes of three validators  (delimiter “,”). Can include your own public key if desired. |

Notes:

On the front end, all but two of the input fields are the same (ownershipIdList and validatorList). Much of the code can be recycled to save time.

On the back end, we use the same ABI for the deployed MyGatekeeper, but the address of MyGatekeeper is used to interact with the users contract.

One thing that needs to be done is to change the contract in MyGatekeeper.sol and Gatekeeper.sol to create a MyGatekeeper.sol contract for the user, rather than coidGateKeeper.sol

A second note is that we cannot instantiate an event listener object outside of the post request, as the continuous event listening depends on the contract. For this reason, the event listener currently object is instantiated in the post request, for each contract address. It remains to verify that this object does not expire upon termination of the post request; if this is the case, we will have to make a global list that holds the event listener objects, and put them in this list at each post request. This will make it continuous.

Should the notification endpoints change? If so, they will need to be added in index.js and in theNotifier.