q

**Bot Creation**

***Step I***

Write a function in botcommonfunction.ts of gibots-node-api.

Input = {

 from : ‘’,

 to : ‘’

}

Output = {

 success : ‘’

}

 // {} , {}, 0, str/objId, 0

 async power(input, outPut, botId, projectId, iterationId) {

 try {

 let num = input.num; // 4;

 let powerVal = input.raiseTo; // 3

 let finalpower = 1;

 while(powerVal--)

 {

 finalpower = finalpower \* num;

 }

 outPut['result'] = finalpower;

 return { message: '', status: 0, data: outPut };

 } catch (e) {

 console.log('bot2 Error', e)

 return { message: '', status: 1, data: e };

 }

 }

***Step II***

Insert function name in botfuncitons collection

db.getCollection("botfunctions").insert({

 "isDeleted": false,

 "functionName": "power"

})

***Step III***

Create bot from frontend, add input output parameters which are used in code



**Process Creation**

***Step I***

First add Start and end bots by dragging from the upper screen, and then add ur bots orderwise.

******

***Step II***

After that, we have 2 options to pass input to the bots.

One is through Bot Details ,



Second is through open mapping where we can access

all variables ( added while creating process ) + outputparameters of the exact previous bot

And map these to our bots input



OutputWrite

Here we can assign/reassign the variables with the current bots output.

let a = 0;

{ Bot run

 b = 11; }

[ Outwrite a = b] //a = 11

.

.

.

is bot use karna hai ( open mapping a)

.

{ Bot run c = 0;} // a = 0

[ Outwrite a = c]



***Step III***

Saving the process



Can give access to particular user and role with permissions for that process



**Collections and other**

In DB, these all processes are saved in these collections: - processes and projects

db.processes.find({

 processName : /Sahas Power/i

})

db.projects.find({

 name : /Sahas Power/i

})

 "successor" : [ Double("1713936136127") ], // which bot to run next

 "andPredecessor" : [ Double("1713936132058") ], /which bot is before this

 "orPredecessor" : [ ],

events

Whenever we trigger the process one entry which contains the metadata of process which we triggered in events collection.

db.events.find({

 \_id : ObjectId("6628a0735ac40e1f0ee68175")

})

Eventstatues

Here all bots have separate document with their prior status and input outputparameters.

db.eventstatuses.find({

 eventId : ObjectId("6628a0735ac40e1f0ee68175")

})

