Any request from the client will have to include an API key

## From client

GET master/job/passcode – reply with job process hostname, port and temporary access token (maybe)

POST master/job – to create a job (master will reply with new passcode)

DELETE master/job/passcode

GET worker job process/job/passcode

GET worker job process/job/passcode/results

PATCH worker job process/job/passcode – to edit job parameters

## From master

GET worker monitor process/available\_worker

GET worker/job/passcode

POST worker/job

PATCH worker/job/passcode

DELETE worker/job/passcode

## From worker monitor process

GET worker/status

## From worker

POST worker monitor process/worker/id - on worker start-up

PATCH worker monitor process/worker/id - if worker status changes - i.e. empty or full

PATCH master/job/passcode - if job info changes

## From worker job process

PATCH worker/job/passcode - if job info changes

# Resources

## Worker (on worker process monitor)

Hostname

Port

“Resource Load”: {

“timestamp1”: {

“cpu”; xx,

“memory”; xx…

},

“timestamp2”: {

“cpu”; xx,

“memory”; xx…

}

}

## Job (on master, worker, job process)

“Passcode”: {

“Job Type”: “string”

“Job Params” : {

"some\_job\_param1": "string",

"some\_job\_param2": …

},

“Job Status”: “string”,

“Start-date”: “timestamp”,

“End-date”: “timestamp”,

## Job result (on job process)

“Job-results”: {

“some\_result1”: “string”

}

}

## Job Skeleton

<http://json-schema.org/>

<https://github.com/brutusin/json-forms> - no library dependencies

<http://schemaform.io/> - uses AngularJS

## Users (on master)

“API Key”: {

“User’s Name”: “string”

“Allowed jobs”: {…

# Other links

<https://www.duckdns.org/spec.jsp>

<https://github.com/shred/acme4j>

<http://www.restapitutorial.com/lessons/restquicktips.html>