DOCUMENT : SCL IPFS API

STATUS : DRAFT

VERSON : 0.0.3

1. CREATE SERVER

endpoint: /api/ipfs/server

method: POST

request body:

directory : string : required : fully qualified directory path

tag : string : optional

swarm\_port : integer : optional

api\_port : integer : optional

gateway\_port : integer : optional

max\_dir\_size : integer : optional : this will be in terms of gigabytes : defaults to 100

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {

"id": 1,

"directory": "/home/temp/my\_projects/scl/ipfs\_system/tests/dir1",

"tag": "",

"swarm\_port": 4002,

"api\_port": 5002,

"gateway\_port": 8082,

"max\_dir\_size": 50,

"peer\_id": "QmbCfVUwpMA2an8Y71pDy9yCZonv7mHaHfBcmeKeoe3p3N"

}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

2. DELETE SERVER

endpoint: /api/ipfs/server/<id>

method: DELETE

params:

id : integer : required

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

3. UPDATE SERVER

endpoint: /api/ipfs/server/<id>

method: PUT

params:

id : integer : required

request body:

tag : string : optional

swarm\_port : integer : optional

api\_port : integer : optional

gateway\_port : integer : optional

max\_dir\_size : integer : this will be in terms of gigabytes

is\_private : integer : 0 (False) 1 (True)

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {

"id": 1,

"directory": "/home/temp/my\_projects/scl/ipfs\_system/tests/dir1",

"tag": "avengers",

"swarm\_port": 4002,

"api\_port": 5002,

"gateway\_port": 8082,

"max\_dir\_size": 50,

"peer\_id": "QmbCfVUwpMA2an8Y71pDy9yCZonv7mHaHfBcmeKeoe3p3N",

"pid": 6185

}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

4. GET SERVERS

endpoint: /api/ipfs/server/all

method: GET

params:

page : integer : optional : default 1

records : integer : optional : default 10

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": [

{

"id": 1,

"directory": "/home/temp/my\_projects/scl/ipfs\_system/tests/dir1",

"tag": "",

"swarm\_port": 4002,

"api\_port": 5002,

"gateway\_port": 8082,

"max\_dir\_size": 50,

"peer\_id": "QmbCfVUwpMA2an8Y71pDy9yCZonv7mHaHfBcmeKeoe3p3N"

}

]

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

5. START SERVER

endpoint: /api/ipfs/server/<id>/start

method: PUT

params:

id : integer : required

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {

"pid": 6146

}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

6. STOP SERVER

endpoint: /api/ipfs/server/<id>/stop

method: PUT

params:

id : integer : required

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

7. RESTART SERVER

endpoint: /api/ipfs/server/<id>/restart

method: PUT

params:

id : integer : required

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {

"pid": 7092

}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

8. UPLOAD FILE

endpoint: /api/ipfs/file

method: POST

request body:

fqfp : string : required : fully qualified file path

server\_id : integer : required

tag : string : optional

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {

"id": {

"id": 1,

"fqfp": "/home/temp/my\_projects/scl/ipfs\_system/tests/test\_image.png",

"filename": "test\_image.png",

"tag": "",

"server\_id": 0,

"size": 116752,

"ipfs\_hash": "",

"ipns\_key\_id": 0,

"upload\_status": 0,

"is\_active": 1

}

}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

9. DELETE FILE

endpoint: api/ipfs/file/<id>

method: DELETE

params:

id : integer

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

10. GET FILES

endpoint: /api/ipfs/file/all

method: GET

params:

page : integer : optional : default 1

records : integer : optional : default 10

[SUCCESS]

{

"status": "SUCCESS",

"data": [

{

"id": 1,

"fqfp": "/home/temp/my\_projects/scl/ipfs\_system/tests/test\_image.png"",

"filename": "test\_image.png",

"tag": "",

"server\_id": 1,

"size": 116752,

"ipfs\_hash": "QmPzkPCUCSzTNwWmDhZTh3vE2oK4NwxATF1j88ofn6eyFc",

"ipns\_key\_id": 0,

"upload\_status": 2,

"is\_active": 1

}

]

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

11. PING SERVER

endpoint: /api/ipfs/server/<id>/stop

method: GET

params:

id : integer : required

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

12. CREATE IPNS KEY

endpoint: /api/ipns/key

method: POST

params:

key : string : required

server\_id : integer : required

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {

"id": 1,

"key": "doctorstrange",

"ipns\_hash": "QmVy5CGft1ujZhCV1w9vujGbSddWSpaREnk9EgXLYGrwVH",

"server\_id": 1,

"is\_active": 1

}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

13. PUBLISH KEY

endpoint: /api/ipns/publish

method: POST

params:

key\_id : integer : record id of key

file\_id : integer : record id of uploaded file

ttl = integer : number of hours that IPNS hash is resolvable to the IPFS hash

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {

"id": 1,

"fqfp": "\/home\/crowdmetric-ivar\/my\_projects\/scl\/ipfs\_system\/tests\/test\_image.png",

"filename": "test\_image.png",

"tag": "",

"server\_id": 1,

"size": 116752,

"ipfs\_hash": "QmPzkPCUCSzTNwWmDhZTh3vE2oK4NwxATF1j88ofn6eyFc",

"ipns\_key\_id": 1,

"upload\_status": 2,

"is\_active": 1,

"ipns\_info": {

"id": 1,

"key": "doctorstrange",

"ipns\_hash": "QmVy5CGft1ujZhCV1w9vujGbSddWSpaREnk9EgXLYGrwVH",

"server\_id": 1,

"is\_active": 1

}

}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

14. DELETE KEY

endpoint: /api/ipns/key/<id>

method: DELETE

params:

id : integer

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

14. GET KEYS

endpoint: /api/ipns/key/all

method: GET

params:

page : integer : optional : default 1

records : integer : optional : default 10

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": [

{

"id": 1,

"key": "doctorstrange",

"ipns\_hash": "QmVy5CGft1ujZhCV1w9vujGbSddWSpaREnk9EgXLYGrwVH",

"server\_id": 1,

"is\_active": 1

}

]

}

[FAIL]

{

"status": "FAILED",

"data": {}

}

14. TEST SERVER STORAGE AVAILABILITY

endpoint: /api/ipfs/server/<id>/test

method: GET

params:

id : integer : required : this is the server\_id to test if a file

fqfp : string : required : fully qualified file path

response:

[SUCCESS]

{

"status": "SUCCESS",

"data": {}

}

[FAIL]

{

"status": "FAILED",

"data": {}

}