

pysync.42.fr

API json - Version 1

Préambule

Configuration générale

HOST = "pysync.42.fr:8000"

VERSION = "v1"

PASSWD = "p@ssw0rd"

Requêtes

Les requêtes doivent se faire en https uniquement.

Tous les paramètres doivent être encodés en UTF-8.

Vous devez connaître l'HOST et la VERSION de l'API utilisée, ainsi que le PASSWD d'accès au serveur.

<https://HOST/api/VERSION/>

Les requêtes sont soit en GET, soit en POST, soit potentiellement les 2.

Authentification

Tous les accès au serveur central doivent être mentionner le mot de passe du serveur, ainsi qu'une clé SLAVE et/ou MASTER

https://HOST/api/VERSION/COMMAND?m_key=MASTER&s_key=SLAVE

Algo fonctionnel d'un client

```
conf = read_config("config.json");
auth = conf.server_password;

while (true)
{
    foreach (conf.folders as folder)
    {
        if (!isset(folder.client_id))
        {
            folder.client_id = register_client(auth, folder.s_key, folder.baseurl);
        }
        master_list = get_list(auth, folder.s_key);
        local_list = build_local_list(folder.path);
        diff = diff_list(master_list, local_list);
        if (folder.m_key == "") // not a master on this folder
        {
            foreach (diff as file)
            {
                urls = get_file(auth, folder.s_key, file.path);
                if (dl_file(urls[0], folder.path) == true)
                {
                    put_list(auth, folder.m_key, folder.s_key, Array(file));
                }
            }
        }
        else // the master doesn't need to dl anything, just publish his file list
        {
            // remove file not on master anymore
            del_list(auth, folder.m_key, folder.s_key, diff);
            diff = diff_list(local_list, master_list); // reverse diff
            put_list(auth, folder.m_key, folder.s_key, diff);
        }
    }
}
```

Commandes

REGISTER_CLIENT: publier son baseurl et récupérer son client_id

Cette étape est nécessaire pour pouvoir publier sa liste de fichiers avec PUT_LIST

Requête:

- commande: register_client
- méthode: GET

Paramètres requis:

- auth: le mot de passe du serveur
- s_key: la cle slave
- baseurl: l'url racine permettant d'accéder aux fichiers de ce folder

Paramètres optionnel:

- m_key: la clé master

Réponse:

- success: TRUE si la requête a réussi, sinon FALSE
- error: le détail de l'erreur en cas d'échec
- data.client_id: l'id à fournir pour les requêtes put_list

Exemple:

`http://localhost:8000/api/v1/register_client?s_key=slave_secret&auth=ChangeMe&baseurl=http://toto.42.fr/`

réponse:

```
{
  "success": true,
  "data": {
    "client_id": "42"
  }
}
```

GET_LIST: obtenir la liste des fichiers correspondant à une SLAVE key

Requête:

- commande: get_list
- méthode: GET ou POST

Paramètres requis:

- s_key: la cle slave
- auth: le mot de passe du serveur

Réponse:

- success: TRUE si la requête a réussi, sinon FALSE
- error: le détail de l'erreur en cas d'échec
- data: un tableau des fichiers à synchroniser comportant pour chacun les champs:
 - name: le chemin complet du fichier
 - size: sa taille en octets
 - mtime: sa date de dernière modification
 - md5: le hash md5 du contenu du fichier

Exemple:

https://HOST/api/VERSION/get_list?s_key=SLAVE&auth=PASSWD

http://localhost:8000/api/v1/get_list?s_key=slave_secret&auth=ChangeMe

réponse:

```
{
  "succes": true,
  "data": [
    {
      "id": 1,
      "path": "toto.txt",
      "size": 42,
      "mtime": "2015-12-01T16:19:35.000Z",
      "md5": "8e8c20055a1c981eaceb45526b461159"
    },
    {
      "id": 2,
      "path": "config.json",
      "size": 100,
      "mtime": "2015-12-01T16:27:50.000Z",
      "md5": "06b2d3b23dce96e1619d2b53d6c947ec"
    }
  ]
}
```

GET_FILE: obtenir les adresses d'un fichier correspondant à une SLAVE key et un PATH

Requête:

- commande: get_file
- méthode: GET ou POST

Paramètres requis:

- s_key: la cle slave
- auth: le mot de passe du serveur
- path: le chemin complet du fichier

Réponse:

- success: TRUE si la requête a réussi, sinon FALSE
- error: le détail de l'erreur en cas d'échec
- data: un tableau des fichiers à synchroniser comportant pour chacun les champs:
 - name: le chemin complet du fichier
 - size: sa taille en octets
 - mtime: sa date de dernière modification
 - md5: le hash md5 du contenu du fichier
 - baseurl: l'adresse url racine où se trouve le fichier

Exemple:

https://HOST/api/VERSION/get_list?s_key=SLAVE&auth=PASSWD

http://localhost:8000/api/v1/get_file?s_key=slave_secret&auth=ChangeMe&path=toto.txt

réponse:

```
{
  "succes": true,
  "data": [
    {
      "id": 1,
      "path": "toto.txt",
      "size": 42,
      "mtime": "2015-12-01T16:19:35.000Z",
      "md5": "8e8c20055a1c981eaceb45526b461159",
      "baseurl": "http://10.42.1.22:8000/test/"
    }
  ]
}
```

PUT_LIST: publier la liste des fichiers locaux correspondant à une SLAVE ou une MASTER key

Requête:

- commande: put_list
- méthode: POST

Paramètres requis:

- s_key: la cle slave
- auth: le mot de passe du serveur
- data: le tableau au format JSON des fichiers (voir format en dessous):

Paramètres optionnel:

- m_key: la clé master

Réponse:

- success: TRUE si la requête a réussi, sinon FALSE
- error: le détail de l'erreur en cas d'échec

Exemple de tableau des fichiers:

```
[
  {
    path: "toto.txt",
    size: 42,
    mtime: "2015-12-01T16:19:35.000Z",
    md5: "8e8c20055a1c981eaceb45526b461159"
  },
  {
    path: "config.json",
    size: 100,
    mtime: "2015-12-01T16:27:50.000Z",
    md5: "06b2d3b23dce96e1619d2b53d6c947ec"
  }
]
```

```
curl -X POST -d 'data=[{"path":
"toto.txt","size":"42","mtime":"2015-12-01T16:19:35.000Z","md5":
"8e8c20055a1c981eaceb45526b461159"},{"path":"config.json","size":"100","mt
ime":"2015-12-01T16:27:50.000Z","md5":"06b2d3b23dce96e1619d2b53d6c94
7ec"}]'
```

http://localhost:8000/api/v1/put_list?s_key=slave_secret&auth=ChangeMe