L'API *Write and Read* est écrite en Java et nous l'utilisons pour notre module de versionning.

Obtenir les classes

Si vous possédez une copie de EzGED, les jar contenant les classes nécessaires se trouvent dans :

```
C:\nchp\usr\local\nchp\ezged\www3\resources\applet
```

Les fichiers jar dont vous aurez besoin sont :

- ezApplet.jar
- warApplet.jar

Vous pouvez aussi télécharger le jar suivant qui contient toutes les librairies nécessaires :

http://updates.nchp.net/WarAppletAll.jar

Utilisation

Upload simple

Un exemple illustrant comment utiliser l'API pour uploader un fichier sur EzGED.

```
import war.*;

public class TestUpload {
    public static void main(String[] args) {
        String filepath = "C:\\prog\\test\\pdf\\image_vec.pdf";
        String url = "http://localhost/ezged3/data/pupload.php";
        String sessionid = "96745501";
        WarApplet applet = new WarApplet();
        String result = applet.upload(filepath, url, sessionid);
    }
}
```

On compile en ajoutant dans notre classpath le jar qui contient toutes les dépendances :

```
>javac -cp .;WarAppletAll.jar TestUpload.java
```

Puis on exécute :

```
>java -cp .;WarAppletAll.jar TestUpload
```

Ce qui nous donne la sortie suivante avec notre exemple:

```
Applet : upload C:\prog\test\pdf\image_vec.pdf to
http://localhost/ezged3/data/pupload.php
Applet : _upload
C:\Users\Florian\.ezged\image_vec.pdf
CHUNKS:4
chunk 0
{"success":true,"chunkpart":true,"message":"Part file successfully
uploaded.","partPath":"C:\\nchp\\var\\spool\\ezged\\instance\/enqueued\/\\im
age_vec.1423583573.pdf"}
chunk 1
{"success":true,"chunkpart":true,"message":"Part file successfully
uploaded.","partPath":"C:\\nchp\\var\\spool\\ezged\\instance\/enqueued\/\\im
age_vec.1423583574.pdf"}
chunk 2
{"success":true,"chunkpart":true,"message":"Part file successfully
uploaded.","partPath":"C:\\nchp\\var\\spool\\ezged\\instance\/enqueued\/\\im
age_vec.1423583574.pdf"}
chunk 3
{"success":true,"chunkpart":false,"message":"File successfully
uploaded.","filePath":"C:\\nchp\\var\\spool\\ezged\\instance\/enqueued\/\\im
age_vec.1423583574.pdf"}

{"success":true,"chunkpart":false,"message":"File successfully
uploaded.","filePath":"C:\\nchp\\var\\spool\\ezged\\instance\/enqueued\/\\im
age_vec.1423583574.pdf"}
```

Upload vers un COLD

Cet exemple montre comment uploader un ou plusieurs fichiers vers le répertoire d'attente. Ce répertoire doit obligatoirement se situer sous l'arborescence **ocr\wait** (e.g : C:\nchp\var\spool\ezged\instance\ocr\wait)

Le code n'est pas si différent, nous appelons simplement la méthode

```
public String _upload(String filepath, String url, String sessionid, String
waitdir)
```

Plutôt que

```
public String upload(final String filepath, final String url, final String
sessionid)
```

Ce qui nous donne

```
import war.*;

public class TestUpload {
    public static void main(String[] args) {
        String filepath = "C:\\prog\\test\\pdf\\image_vec.pdf";
```

```
        String url = "http://localhost/ezged3/data/pupload.php";
        String sessionid = "96745501";
        WarApplet applet = new WarApplet();
        String result = applet._upload(filepath, url, sessionid, "images");

                //Ici le code pour parser le resultat
    }
}
```

Nous ne décrivons pas ici comment parser le résultat qui est au format JSON. Vous pouvez utiliser la librairie de votre choix.

Voici un exemple de retour au format JSON du script pupload.php :

retour.json

```
{
    "success":true,
    "chunkpart":false,
    "message":"File successfully uploaded.",
"filePath":"C:\\nchp\\var\\spool\\ezged\\instance\/ocr\/wait\/tutoriel_
service_cold\/fichier2.pdf"
}
```

Les fichiers dont la taille excède 1.5Mo seront découpés. Vous aurez donc plusieurs retours pour le même fichier. Tant qu'il ne s'agira pas du fichier complet chunkpart aura la valeur true.

Il faudra donc vous assurer d'avoir le retour suivant pour success et chunkpart :

```
"success":true,
"chunkpart":false,
```

From:
http://wiki.ezdev.fr/ - **EzGED Wiki**

Permanent link:
http://wiki.ezdev.fr/doku.php?id=dev:api:war

Last update: **2023/03/17 09:56**