

## SISTEMAS WEB

### 4ª Práctica: eGela2Dropbox

#### OBJETIVO

La aplicación **eGela2Dropbox** permite subir de manera selectiva los ficheros PDF del aula virtual de la asignatura Sistemas Web en eGela, a un directorio de Dropbox.

Para realizar esta práctica se facilita el esqueleto la aplicación con la interfaz gráfica. El/La alumno/a deberá completar los huecos de código para conseguir el correcto funcionamiento de la misma. Este código se divide en dos partes claramente diferenciadas:

1. **Conexión a eGela.** Rellenar los huecos de *eGela.py* reutilizando código de la práctica 2.
2. **Acceso a Dropbox:** Hay que programar 6 peticiones HTTP en *Dropbox.py* utilizando el API de Dropbox. Estas peticiones implementan la delegación de autenticación y autorización mediante OAuth, así como el listado del contenido de un directorio, la subida y eliminación de ficheros de un directorio, y la creación de un directorio.

En el siguiente vídeo se muestra el funcionamiento de las diferentes funcionalidades de la aplicación: <https://www.youtube.com/watch?v=qTnnSdib8Fc>

#### CALIFICACION:

La calificación de la tarea se divide en dos partes:

- El 80% de la calificación, que se obtiene de que la aplicación funcione correctamente rellenando huecos de *eGela.py* y *Dropbox.py*.
- El 20% de la calificación, que se obtiene de añadir mejoras en la aplicación utilizando el API de Dropbox, incrementando en un 5% por cada funcionalidad adicional.

#### ENTREGABLES

1. **Las clases de la aplicación:** *eGela.py* y *Dropbox.py*. Opcionalmente, se entregará el código de aplicación completa si el/la alumno/a decide añadir alguna funcionalidad que modifique su interfaz.
2. **Video de la ampliación** en el que se muestre el correcto funcionamiento de la aplicación y de las ampliaciones realizadas

## ASOS E INSTRUCCIONES PARA LA PRÁCTICA

1. *Crear un proyecto en PyCharm*
2. *Realizar conexión con eGela*
3. *Conseguir el listado de PDFs del aula virtual*
4. *Descargar un PDF del aula virtual*
5. *Registrar la aplicación en Dropbox*
6. *Delegación de la autenticación y autorización en Dropbox con OAuth*
7. *Trabajando con el API de Dropbox*
8. *Añadir alguna funcionalidad utilizando el API de Dropbox*

### 1.- Crear el proyecto en PyCharm

Se creará un proyecto en PyCharm en el que se instalarán todas las librerías necesarias. Dentro de ese proyecto, se cargarán los ficheros con el esqueleto de la aplicación. Los ficheros de la aplicación están disponibles en un fichero ZIP en el siguiente enlace de eGela:

<https://egela.ehu.eus/mod/resource/view.php?id=6484507>

### 2. Realizar conexión con eGela

Los pasos necesarios para llevar a cabo este apartado se analizaron en la clase de PA de la semana 5 y se utilizaron en la realización de la práctica 2. Partiendo de esta base, se deberán rellenar los huecos indicados en el método:

```
check_credentials(self, username, password, event=None)
```

Este método recibe como parámetros de entrada el nombre usuario y la contraseña de eGela. Al finalizar su ejecución, este método actualiza:

- El atributo **\_login** a 1 si el usuario se ha autenticado correctamente.
- El atributo **\_cookie** con la cookie de sesión en eGela.
- El atributo **\_curso** con la uri del curso de Sistemas web.

### 3. Conseguir el listado de PDFs del aula virtual

Al igual que en el apartado anterior, partiendo del código de la práctica 2., se deberán rellenar los huecos indicados en el método:

```
get_pdf_refs(self)
```

Este método construye y devuelve un array (atributo **\_refs**) que contiene el listado con los pares nombre-uri correspondientes a los nombres de los PDFs y sus URLs de descarga. Por ejemplo:

```
[{'pdf_name': 'Presentacion.pdf', 'pdf_link': 'https://egela.ehu.eus/Presentacion.pdf'},  
 {'pdf_name': 'Planificacion.pdf', 'pdf_link': 'https://egela.ehu.eus/Planificacion.pdf'},  
 {'pdf_name': 'ISoftware.pdf', 'pdf_link': 'https://egela.ehu.eus/Software.pdf'},  
 ...]
```



Como nombre de fichero se debe escoger el nombre del propio fichero no la etiqueta de recurso con el que se visualiza en eGela; por ejemplo:

Nombre del fichero: SW\_2023-01-27\_1\_Instalación\_Software.pdf

Etiqueta del recurso con el que se visualiza en eGela: 2023-01-27 INSTALACIÓN SOFTWARE I

### 3. Descargar un PDF del aula virtual

Partiendo del código de la práctica 2, se deberá rellenar el método:

```
get_pdf(self, selection)
```

A este método se le pasa como parámetro el índice de la entrada a descargar dentro del atributo `_refs`. Y devuelve el nombre del fichero y su contenido.

### 4. Registrar la aplicación en Dropbox

Para poder trabajar con Dropbox es necesario **registrar la aplicación** en Dropbox.

El registro de una aplicación en Dropbox se explica en el siguiente vídeo:

<https://www.youtube.com/watch?v=VJkD6cmpcag>

En el código se indica el valor que debe asignarse al campo **redirect\_uri**.

### 5. Delegación de la autenticación y autorización en Dropbox con OAuth

Los pasos están descritos en la documentación.

Aspectos generales:

<https://developers.dropbox.com/es-es/oauth-guide>

Aspectos concretos (API de Dropbox):

<https://www.dropbox.com/developers/documentation/http/documentation#authorization>

Utilizando esta documentación deberéis completar el método:

```
do_oauth(self)
```

Este método deberá actualizar el atributo `_access_token`.

### 6. Trabajando con el API de Dropbox

La documentación para trabajar con el API de Dropbox está disponible en:

<https://www.dropbox.com/developers/documentation/http/documentation>

Para realizar pruebas con el API de Dropbox, os recomendamos el siguiente procedimiento:

<https://www.youtube.com/watch?v=Ulwsi4z-6QE>

Para generar un **access\_token** para hacer pruebas:

<https://dropbox.github.io/dropbox-api-v2-explorer>



Operaciones que hay implementar en *Dropbox.py*:

- **Listar contenido de una carpeta de Dropbox.** Este método actualiza el listado de ficheros en la interfaz de la aplicación con el contenido de la carpeta cuya ruta se define en el atributo *\_path*.

Consultar:

[https://www.dropbox.com/developers/documentation/http/documentation#files-list\\_folder](https://www.dropbox.com/developers/documentation/http/documentation#files-list_folder)

```
list_folder(self, msg_listbox)
```

La respuesta de la petición a Dropbox, en formato json, se pasará a la función `helper.update_listbox2(msg_listbox, self._path, contenido_json_dict)`

en formato diccionario; es decir:

contenido\_json devuelto por Dropbox → contenido\_json\_dict

- **Subir un fichero a Dropbox.**

Consultar:

<https://www.dropbox.com/developers/documentation/http/documentation#files-upload>

```
transfer_file(self, file_path, file_data)
```

Los parámetros de entrada son:

*file\_path*: nombre del fichero que se corresponde con el del parámetro path indicado en el API de Dropbox

*file\_data*: contenido del fichero.

- **Borrar un fichero a Dropbox.**

Consultar:

<https://www.dropbox.com/developers/documentation/http/documentation#files-delete>

```
delete_file(self, file_path)
```

El parámetro de entrada será:

*file\_path*: nombre del fichero que se corresponde con el del parámetro path indicado en el API de Dropbox.

- **Crear una nueva carpeta a Dropbox.**

Consultar:

[https://www.dropbox.com/developers/documentation/http/documentation#files-create\\_folder](https://www.dropbox.com/developers/documentation/http/documentation#files-create_folder)

```
create_folder(self, path)
```

El parámetro de entrada es:

*file\_path*: nombre del fichero que se corresponde con el del parámetro path indicado en el API de Dropbox.



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea



SISTEMEN INGENIERITZA ETA AUTOMATIKA SAILA  
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

### *8.- Añadir funcionalidad utilizando el API de Dropbox*

Después de analizar la aplicación, proponer e implementar alguna sencilla mejora a la aplicación utilizando alguna de las peticiones HTTP del API de Dropbox.