

Memoria PSI

1. Introduccion

Para desarrollar esta aplicación hemos hecho uso de la arquitectura Django. El nombre de nuestro proyecto es workflowrepository. Nuestro proyecto contiene tres aplicaciones bien definidas: 'find', 'data', 'upload'.

Para hacer uso de estas aplicaciones es necesario que las añadamos al apartado de INSTALLED_APPS dentro del fichero settings.py.

Nuestra aplicación hace uso de una base de datos. Si la variable de entorno SQLITE se encuentra definida haremos uso de una base de datos sqlite y si no esta definida sera de tipo postgres.

Esta arquitectura se basa en MTV (model-template-view), es decir, un modelo en el que se especifican los distintos objetos de la aplicación, los datos de la misma: una vista para representar los modelos y definir las funciones; y los templates para hacer uso de los distintos servicios de la aplicación.

Las diferentes URLs de los servicios estan definidas en el archivo urls.py de workflowrepository, en el cual tambien se añaden las urls de las aplicaciones upload y find.

2. Aplicacion data

Esta aplicación contiene los modelos definidos en la aplicación, WorkFlow y Category. En esta aplicación tambien se definen los diferentes campos de la ventana admin de la aplicación, asi como una serie de comandos para poblar y probar los bases de datos. Ademas de estos tambien contiene un fichero de test para probar la funcionalidad de la aplicación.

Los comandos definidos en esta aplicación se pueden llamar desde el directorio principal con los siguientes comandos:

```
python manage.py query_data  
python manage.py populate
```

3. Aplicacion find

Esta aplicación contiene las vistas relacionadas con el visionado de workflows, listado de workflows y categorias, busqueda de workflows, borrado de workflows, descarga de workflows. Todas estas funcionalidades estan definidas en el fichero views.py. Estas views hacen uso de los templates definidos en la carpeta "static" del directorio principal.

Esta aplicación tambien tiene un fichero de test para probar la funcionalidad.

Para descargar los workflows esta desactivada la funcion de verificacion por reCAPTCHA, ya que los test de descarga no funcionan si esta funcionalidad esta activada. Si se quiere activar esta funcionalidad es necesario descomentar una serie de lineas en la funcion de download del archivo views.py (todos los cambios necesarios estan explicados en este fichero).

Las URLs para hacer uso de estas funcionalidades estan definidas en el fichero urls.py.

4. Aplicacion upload

Esta aplicación contiene la funcionalidad para subir WorkFlows a la aplicación. Para subir workflows es necesario realizar el proceso de validacion por reCAPTCHA. La clave privada esta almacenada en el fichero settings.py.

Para rellenar los campos se hace uso de un formulario definido en el archivo forms.py.

5. Decisiones propias

- Es necesario hacer validacion por reCAPTCHA tanto en el proceso de descarga como en el de eliminar. Para no poner un reCAPTCHA en cada formulario, he creado un formulario unico que hace uso de una funcion de JavaScript, que dependiendo del boton que se pulse se realice una accion u otra.

- He tenido que realizar unos pequeños cambios en los test para adaptarlos a mi aplicación. Estos cambios no afectan a la funcionalidad de los test, ya que solo modifican los nombres de las diferentes llamadas, para adaptarlos a mis definciones en mi aplicación. Todos estos cambios se ven reflejados y explicados en los test.