Django Admin: ampliar sus funcionalidades.

13 de Diciembre de 2016 a las 00:40

Todos los programadores en *Django*, en algún momento, han utilizado el panel de control que proporciona, *Django Admin*. Este panel es sencillo y con funcionalidad limitada, aunque para determinadas tareas cumple su cometido.

Todo proyecto *Django* tiene un fichero *admin.py*, donde podemos personalizar su comportamiento y su funcionalidad, añadiendo diversos elementos como; diversos tipos de búsquedas o añadir nuevas acciones. Pero estas modificaciones también son limitadas.

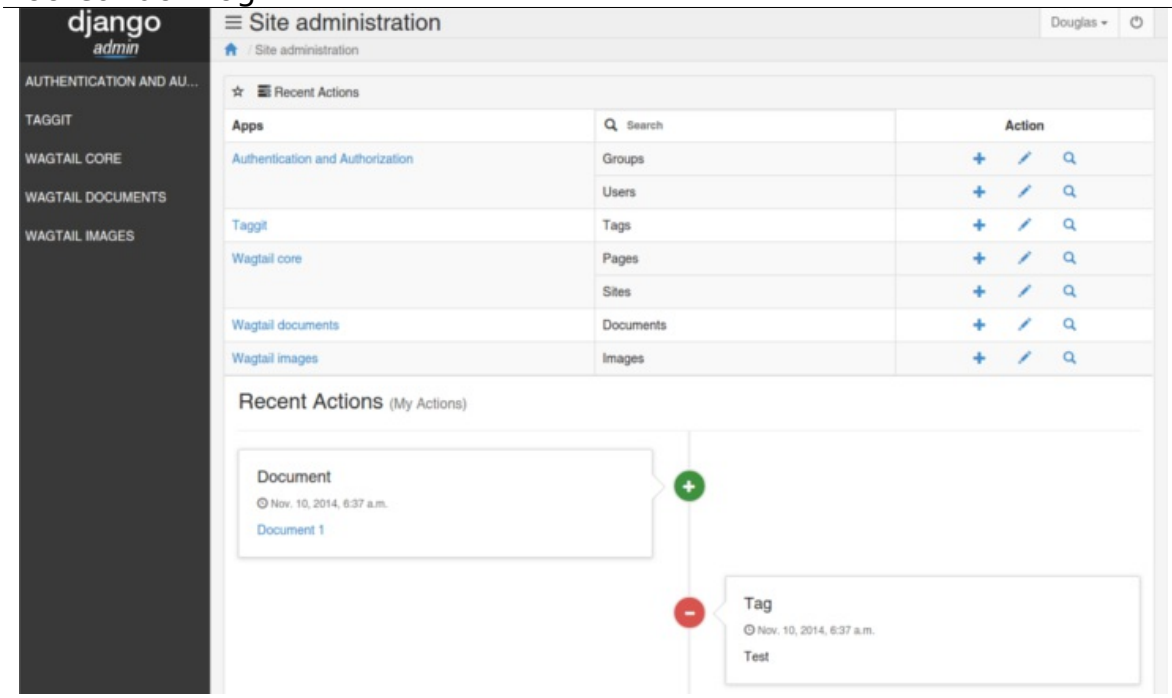
Si queremos añadir mas funcionalidades, podemos modificar el código del panel, algo bastante complejo, o instalar aplicaciones que añadan nuevas funcionalidades al panel. En este artículo veremos una serie de paquetes de *Python* que modificaran *Django Admin*, como bonus un pequeño truco para añadir un cambio en el panel que mejorara la usabilidad.

TEMAS

Existen múltiples temas para modificar el aspecto de *Django Admin*, realmente suelen ser aplicaciones que modifican mas cosas que el aspecto visual, veremos tres de ellas.

Django Bootstrap

Es la interfaz que viene por defecto pero basada en *Bootstrap*, mejorando la visualización en dispositivos móviles, excepto las modificaciones visuales, no trae muchas modificaciones respecto al original.



Su instalación es muy sencilla.

```
pip install bootstrap-admin
```

Añadirlo en el *settings.py* apartado INSTALLED_APPS

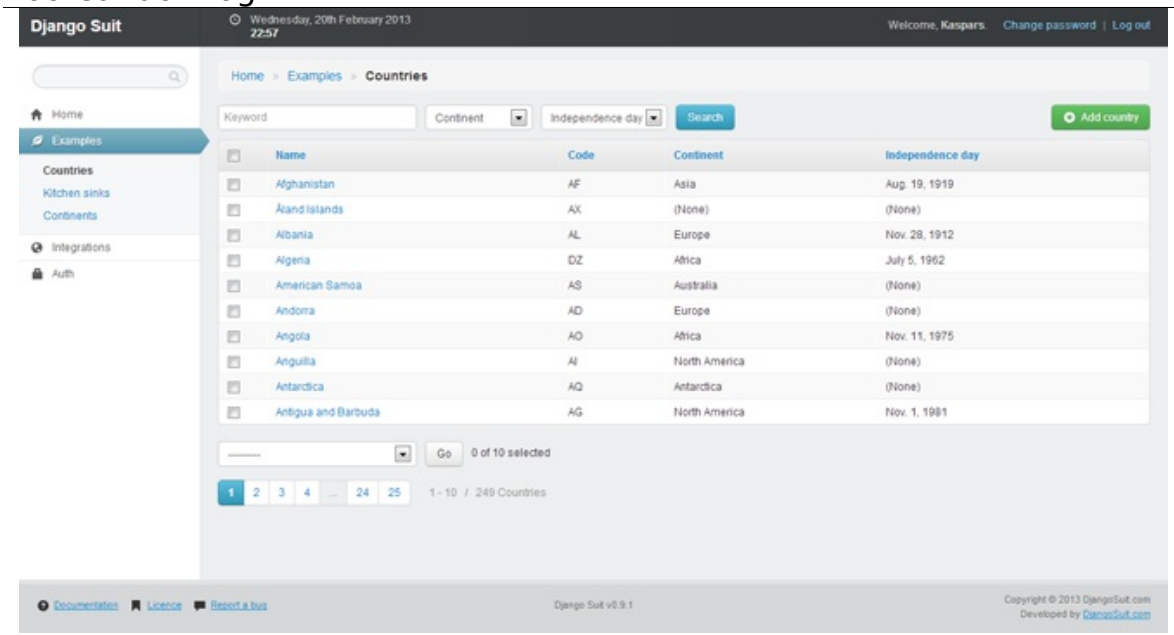
```
INSTALLED_APPS =(
   # ...
   'bootstrap_admin'# always before django.contrib.admin
   'django.contrib.admin',
    # ...
)
```

Para mas información su página de GitHub [django-admin-bootstrap.]

Django Suit

Este tema es comercial, pero dispone de una versión gratuita para un uso no comercial. Integra algunas aplicaciones en el admin y modificaciones visuales que mejoran la usabilidad, como el anterior esta construido sobre *Bootstrap.* Página web http://djangosuit.com/

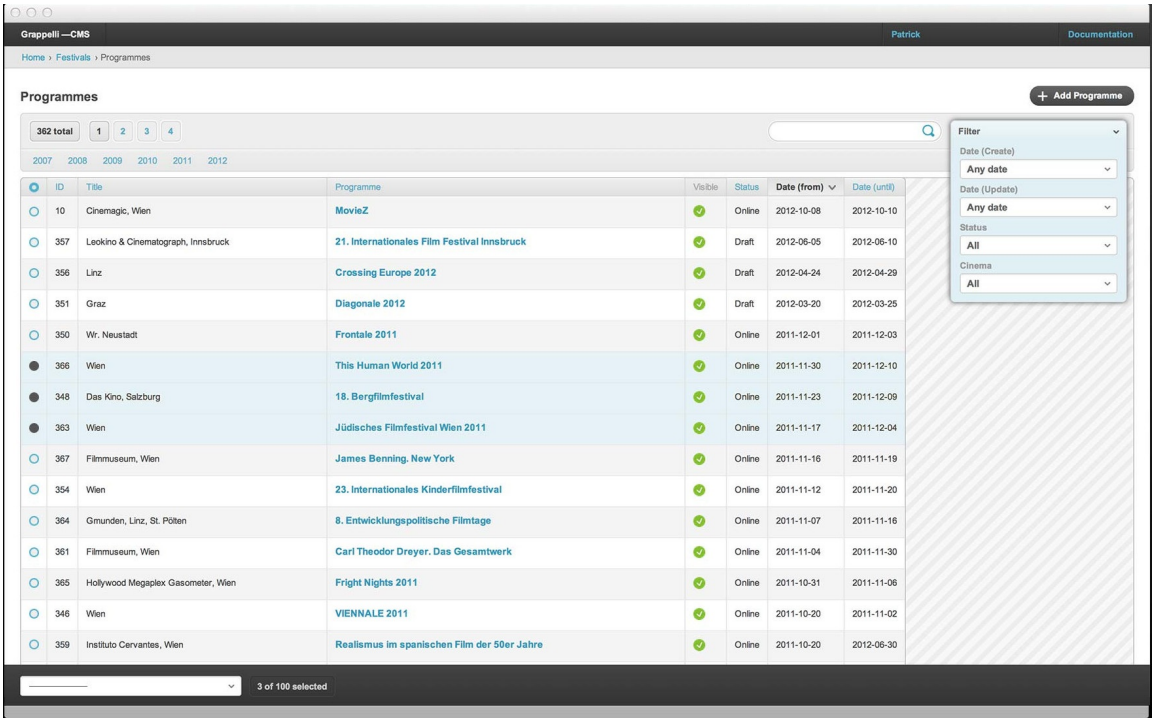
Para la instalación mirar la documentación disponible aquí.



Django Grapelli

Es el que utilizo actualmente, sobrio, configurable y con una buena documentación. Es compatible con otras aplicaciones como; TinyMCE o Django-filebrowser.

Página web http://grappelliproject.com/, documentación (muy recomendable leer) http://django-grappelli.readthedocs.io/en/latest/



APLICACIONES PARA DJANGO

Django Admin es muy bueno para introducir datos en nuestro proyecto Django, para mi es su principal función, pero es limitado en otras características. Para paliar esas

limitaciones, existen un conjunto de aplicaciones que se integran en el *Django Admin* y aumentan su funcionalidad.

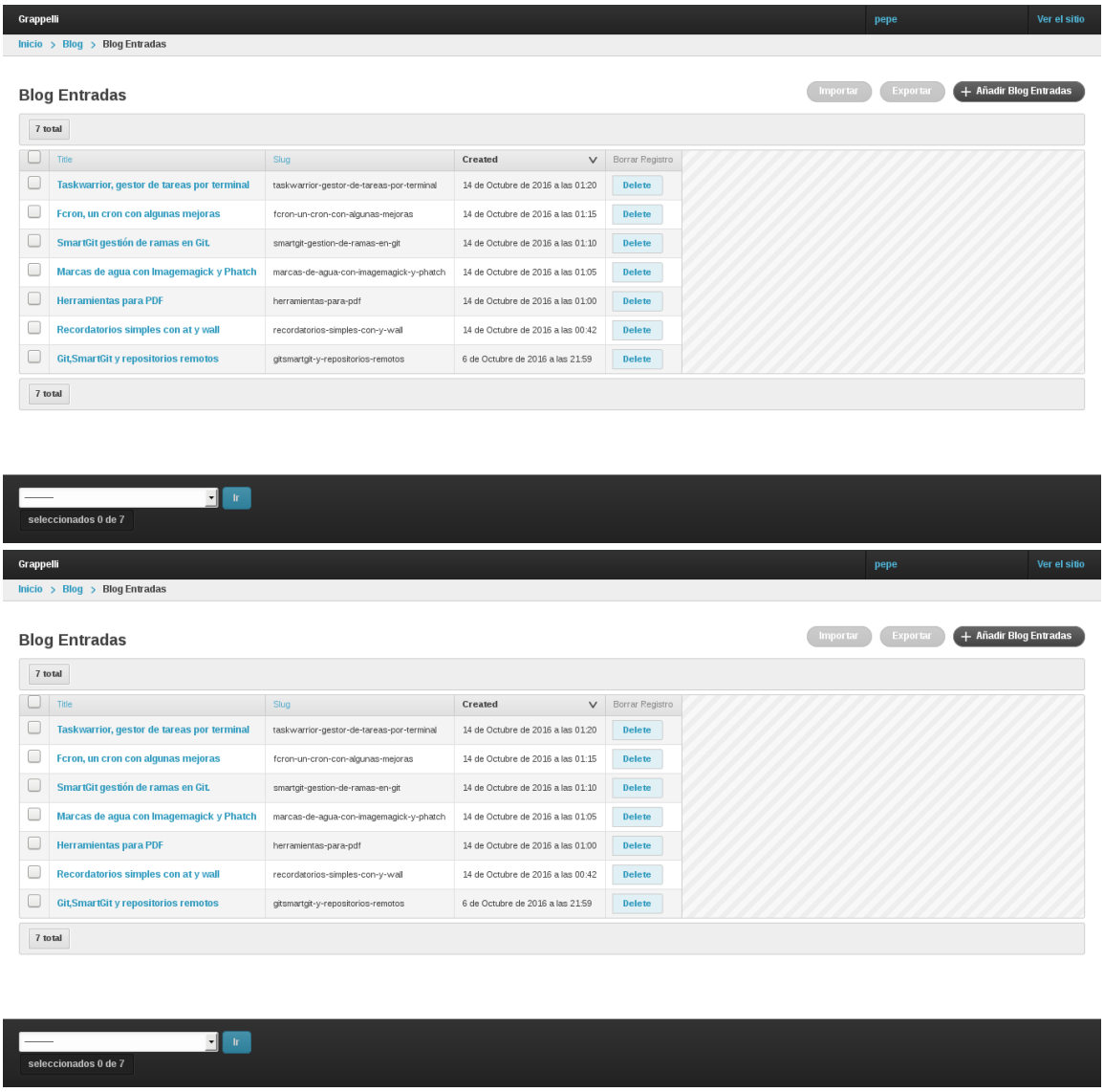
En este artículo presentamos tres aplicaciones.

Django-import-export

Cuando estamos introduciendo datos desde el *Admin* de *Django* no tenemos opción de volcar los datos a un fichero, para tener copia que nos servirá de copia de seguridad. Si perdemos los datos de nuestro proyecto no tenemos la opción de recuperar los datos de un copia realizada anteriormente.

Para solucionar estos limitaciones, tenemos *Django-import-export*, que proporciona la posibilidad de volcar los datos de una tabla del *Admin*, *export*, como recuperar datos de una copia, *import*.

La integración con Django Admin se realiza mediante la inserción de dos botones en la parte superior.



Si pulsamos en el botón de *export,* aparece una nueva ventana donde tendremos que escoger el formato y pulsamos en el botón de *Download*, con esto tendremos una copia de los datos. Para importar datos, pulsamos en botón correspondiente, pulsando *Examinar* podremos escoger la copia de datos que nos interese y especificar el formato de esa copia, pulsamos en *Enviar* y los datos aparecerán en la tabla.

La instalación es muy sencilla, mediante el comando *pip* y añadirla en *setting.py*, para mas información mirar la documentación,aquí.

Nota.

Si en Django Admin esta instalado Grapelli, este paquete no mostrara los botones import/export correctamente. Esto es debido a que el paquete utiliza sus propias template que no se integran bien con Grapelli. Para solucionar esto,hay que modificar ciertos ficheros, esta muy bien explicado en el siguiente enlace.

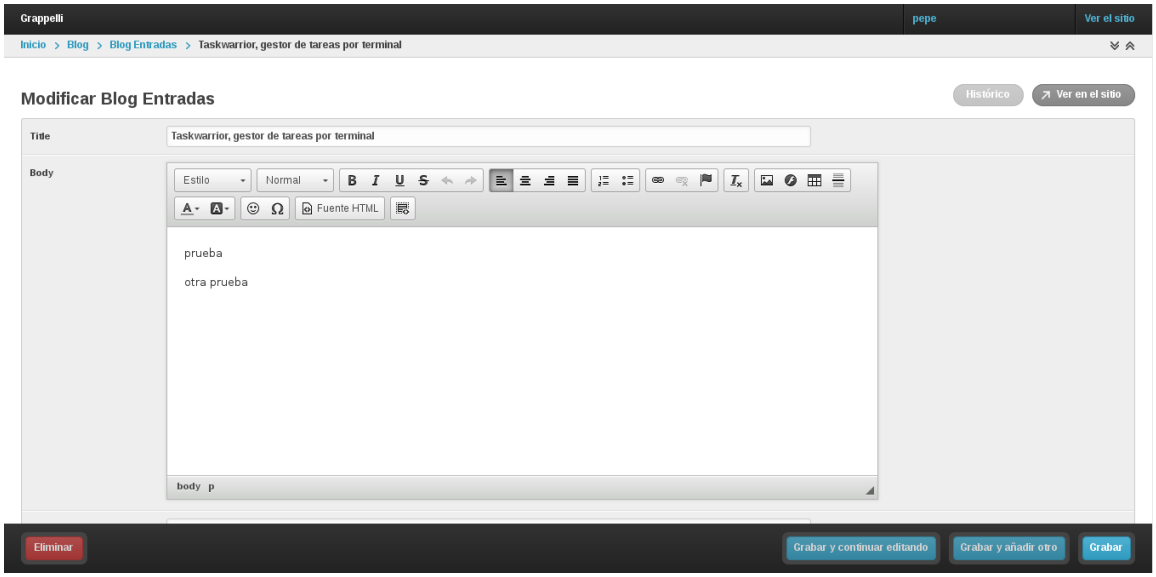
https://github.com/sehmaschine/django-grappelli/issues/435

Fin Nota.

Django-reversion

Esta aplicación proporciona un control de versiones para las instancias de los modelos

La integración con *Django Admin* es mediante la inserción de un botón *Histórico* en la pantalla para insertar datos.



Cada vez que pulsemos en grabar se creara una entrada en el histórico, si pulsamos en el botón *Histórico* mostrara una serie de entradas con las modificaciones que se han realizado, pudiendo volver a estados anteriores.

Django-FSM

Esta aplicación permite declara estados en tus modelos, permite implementar flujos de trabajos en *Django*. Para mas información sobre flujos de trabajo, aquí.

Un ejemplo típico es cuando desarrollamos un blog, un usuario escribe un post y se le asigna un estado de *borrador,* cuando termina lo pública y el post cambia a un estado de *publicado*. Los estados pueden ser configurados según los requerimientos, para mas información página web de *Django FSM.*

Existe un paquete para integrarlo en el *Django Admin*, *Django-fsm-admin*, que facilitará el uso de *Django FSM*.

Un pequeño truco

En *Django Admin* para borrar un registro requiere un proceso en dos pasos, primero seleccionar los registros a borrar y segundo seleccionar la acción de borrar en la parte inferior del admin. Si desea borrar un solo registro es un proceso un poco largo, encontré un pequeño código que inserta botones *Delete* al final de cada registro, no encuentro el enlace donde estaba el código.

Dentro del fichero *admin.py* insertar el siguiente código en la clase.

```
class MyAdmin(ModelAdmin):
  def delete(self,obj):
    return '<input type="button" value="Borrar" onclick="location.href=\'{0}/delete/\'" />'.format(obj.pk)
```

Por último, recordar que he creado un canal de Telegram Un Día Una Aplicación https://telegram.me/UnDiaUnaAplicacion, donde cada dái muestra una aplicación, servicio o comando relacionado con Linux. Si te interesa no dudes en unirte.