**Manual de desarrollo: DYNAMIC FORMS**

**Introducción**

Dynamic Forms es un framework desarrollado en Python-Django pensado para la generación de formularios dinámicos multi-página y multi-caminos. La creación de dichos formularios se realiza mediante una aplicación web con una interfaz amigable la cual se ha probado que funciona correctamente para las últimas versiones de los navegadores Chrome, Mozilla Firefox e Internet Explorer.

**Características y funcionalidades**

Dynamic Forms provee una serie de funcionalidades y características útiles para el desarrollo de formularios con múltiples propósitos. Entre esas características se encuentran:

* Herramienta para la creación, edición y modificación de formularios.
* Posibilidad de mantener distintas versiones de un mismo formulario.
* Interfaz amigable y responsive para el completado de los formularios.
* Posibilidad de generar formularios multi-página.
* Múltiples tipos de campo entre los que se encuentran:
* Numérico
* Línea de Texto
* Área de Texto
* Lista Simple Response
* Lista Multiple Response
* E-mail
* Mapa
* Archivo
* Posibilidad de mostrar o ocultar un campo del formulario dependiendo de la respuesta de otros campos.
* Posibilidad de mostrar o ocultar una página del formulario dependiendo de la respuesta de determinados campos.
* Posibilidad de determinar validaciones sobre diferentes tipos de campo.
* Extracción de estadísticas sobre las respuestas de los distintos formularios.
* Exportación de las respuestas a formato CSV.
* Exportación de estadísticas en formato PDF.
* Inclusión de las encuestas como plug-in de Django-CMS.
* QuerySets para el análisis de las respuestas.
* Manejo de Django-Signals ante la ocurrencia de determinados eventos.
* Posibilidades de determinar diferentes acciones una vez completado el formulario.

**Requerimientos**

Python 3.4

Django 1.6

AngularJS v1.3.0

**Cómo instalarlo**

Incluir Dynamic Forms a otro proyecto Django es muy sencillo. Simplemente se debe descargar el framework desde los repositorios de python mediante el comando:

pip install dynamicForms

Agregar "dynamicForms" a las INSTALLED\_APPS del setting del proyecto.

INSTALLED\_APPS = (

# other apps

'dynamicForms',

)

En el archivo urls.py del proyecto:

# other imports

import dynamicForms

urlpatterns = patterns('',

# other patterns

url(r'^<url\_base>/', include('dynamicForms.urls'), name='base'),

)

Para poder utilizar el servicio de mails que brinda la aplicación, se debe realizar las siguientes configuraciones en el archivo settings.py del proyecto:

EMAIL\_HOST = <'SERVIDOR\_DE\_CORREO'>

EMAIL\_HOST\_USER = <'CUENTA\_CORREO'>

EMAIL\_HOST\_PASSWORD = <'CONTRASEÑA\_CORREO'>

EMAIL\_PORT = <PUERTO>

EMAIL\_USE\_TLS = True

**Creación de un nuevo formulario**

La creación de un formulario se realiza de manera muy sencilla. Simplemente se deben seguir los siguientes pasos:

1. Dirigirse a la URL base definida y loguearse con el usuario de Django admin.
2. Una vez logueado, presionar el botón de 'New Form' ubicado en el menú izquierdo. La aplicación redirigirá a la pantalla de edición.
3. En la pantalla de edición se deberá seleccionar el titulo con el que se pretende que el formulario sea publicado. Dicho título generara un slug el cual será el identificador del formulario dentro del sistema.
4. En dicha pantalla se podrán realizar todas las modificaciones necesarias para crear el formulario deseado. Para ello se cuenta con una paleta de opciones en el sector derecho de la pantalla. Dicha paleta contiene 3 pestañas.

La primera es la denominada "Palette" y brinda la posibilidad de insertar nuevos campos y páginas al formulario.

La segunda pestaña es la de propiedades de campo. En esta pestaña se podrán editar todas las propiedades del campo que se encuentre seleccionado. Las propiedades varían dependiendo del tipo de campo que se esté editando.

Por último, la tercer pestaña es la encargada de las configuraciones. Entre sus opciones se encuentran la de seleccionar si utilizar captcha o no, seleccionar las acciones a realizar una vez completado el formulario y configurar la lógica de mostrar u ocultar campos o páginas.

1. Una vez realizada todas las modificaciones deseadas al formulario se presiona el botón 'Save' o 'Publish'.
2. En la pantalla principal de la aplicación se podrá ver en la lista de formularios el nuevo formulario creado.

**Estados de un formulario**

Los formularios generados mediante la herramienta tienen diferentes versiones. Cada una de esas versiones se encuentra en alguno de los siguientes estados.

* Draft: Formulario en estado de edición. No puede ser completado, solamente está permitido ingresar al editor y modificarlo.
* Published: Versión que se encuentra en este momento publicado. No se puede editar pero sí se puede completar. Además tiene permitido la posibilidad de mirar sus respuestas y estadísticas.
* Expired: Versión que estuvo publicada pero que fue sustituida por una nueva versión publicada. Solamente se pueden extraer estadísticas del mismo.

**Completar un Formulario**

Para poder completar un formulario es necesario disponer del mismo en una página web. Actualmente se disponen de 4 maneras de disponer del formulario para completarlo.

* URL:

Se puede mapear un formulario determinado a una URL cualquiera usando la vista "render\_form" .

En el archivo urls.py del proyecto agregar la siguiente línea:

url(r'^<url\_destino>/$','dynamicForms.views.render\_forms',{'instance':'<slug\_del\_formulaio>'}),

* TAG:

Se dispone de un tag de Django el cual permite incluir un formulario dentro de un template preexistente. Es necesario cargar el tag antes de invocarlo. Esto se realiza de la siguiente forma:

{%load visor\_tag%}

Luego el tag se utiliza de la siguiente forma:

{%visor\_template\_tag "<slug\_del\_formulario>" %}

* CMS:

Se creó un plug-in para embeber un formulario de dynamicForms en Django-CMS.

**Restricciones de dynamicForms**

Existen un par de restricciones al momento de utilizar dynamicForms. La primera de ellas se encuentra en que la lógica de bifurcación, entre los campos presenta solo unos pequeños controles de consistencia. Queda en responsabilidad del usuario que genera los formularios la consistencia de la misma. La segunda limitante consiste en la incapacidad de colocar dos formularios en la misma página. Este punto se debe las limitantes que tienen las tecnologías utilizadas.