**TUTORIAL DJANGO**

**INSTALACIÓN**

Primero deberás de instalar python3 y django.

Para installar python 3 recomiendo instalar anaconda: [https://www.anaconda.com/download/#windows](https://www.anaconda.com/download/" \l "windows)

Durante la instalación no olvides de seleccionar el casillero donde se dice para instalarlo en el path.

Luego, deberás de instalar django. Para esto, deberás de ir a “command shell” y poner:

pip install django

Esto instalará django.

**CREACIÓN DE UN PROYECTO**

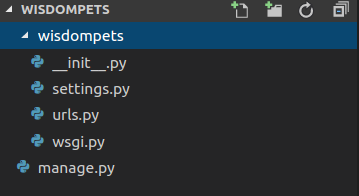
Primero en “command shell” ubicarte en donde quieres crear el proyecto. Cuando crees el proyecto crearás una carpeta. Por ejemplo, si queremos que el nombre del proyecto sea “wisdompets” deberás de poner en el command shell:

django-admin.py startproject wisdompets

Esto creará una nueva carpeta llamada wisdompets en la ubicación donde te encuentras. De no funcionar, prueba escribiendo

django-admin startproject wisdompets

A veces, no debes de poner el “.py” esto depende de multiples factores, el sistema operativo, la versión de windows que uses, la versión de python que uses, entre otros.



Cuando crees el proyecto se creará una carpeta llamada wisdompets, dentro de esta carpeta se crearán distintos archivos “.py”.

Una vez creado el proyecto puedes probar que todo funciones entrando a la carpeta creada en “command shell” con (en este caso, cd wisdompets), dentro de la carpeta del proyecto pon:

python manage.py runserver

Esto abrirá un puerto, normalmente el 8000. Así que anda a un navegador web y pon

localhost:8000

Si todo anda bien podrás entrar a una pantalla de bienvenida de django

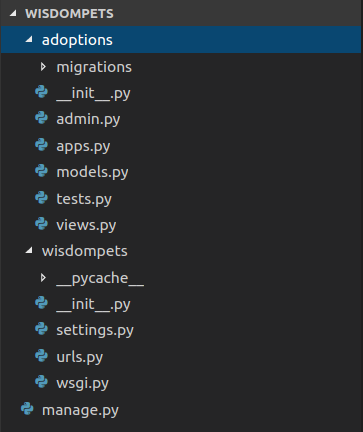
**DESARROLLO DEL PROYECTO**

El primer paso para desarrollar tu proyecto será la creación de un App. Un proyecto puede tener varias Apps. De esta forma se puede organizar el proyecto. Para crear el App, dentro del “command shell” dentro de la ubicación del proyecto pon:

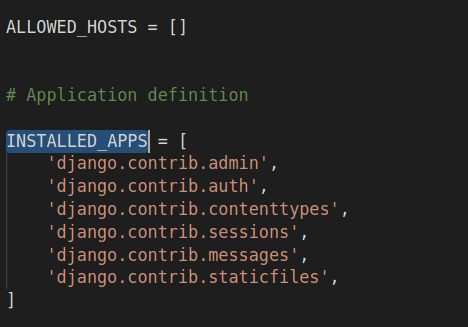
python manage.py startapp adoptions

Este comando creará una nueva carpeta dentro de la carpeta de tu proyecto de nombre “adoptions”. Este es el nombre del App.

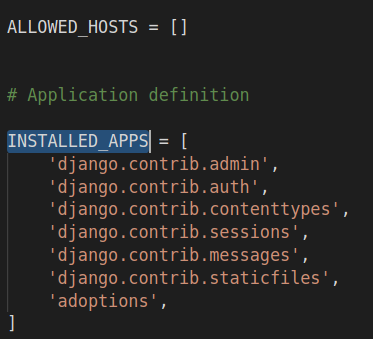
Esta carpeta también tendrá distintos archivos “.py”



Para hacerle saber a “django” que has creado un nuevo App, dentro de la carpeta wisdompets, en el archivo “settings.py” deberás de indicar el nombre del App que has creado. Para esto, dentro de la lista “INSTALLED\_APPS” deberás de incluirlo.



Deberás de poner, al final de la lista la nueva app “adoptions”



**CREACIÓN DEL MODELO DE DATOS**

El modelo de datos describe los datos que se usarán en la aplicación, en este caso la aplicación mostrará información sobre mascotas y sobre las vacunas de las mascotas.

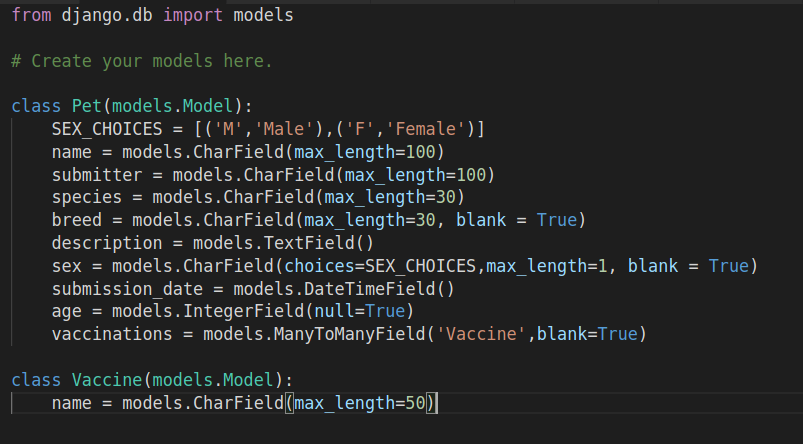
Las variables que tendrás las mascotas serán las siguientes:

|  |  |  |
| --- | --- | --- |
| **Variable** | **Descripción** | **Tipo** |
| name | Nombre de la mascota | CharField |
| submitter | Persona que está dando en adopción la mascota | CharField |
| species | Especie de la mascota (perro, gato) | CharField |
| breed | Raza de la mascota. Puede no tener raza | CharField |
| description | Descripción de la mascota en adopción | TextField |
| sex | Sexo de la mascota (Puede ser nulo | CharField  With choices |
| submission\_date | Fecha que se registro la mascota | DateTimeField |
| age | Edad de la mascota | IntegerField |
| vaccinations | Vacunas que tiene la mascota | ManyToManyField |

El modelo de vacunas, solo tendrá una variable que será el nombre.

|  |  |  |
| --- | --- | --- |
| **Variable** | **Descripción** | **Tipo** |
| Name | Nombre de la vacuna | CharField |

Para crear el modelo de datos dentro de django deberás de ir a la carpeta donde se encuentra en App *adoptions* y deberás modificar el archivo “models.py”



Luego de añadir el código del modelo de datos al archivo “models.py” deberás “decirle” a django que incluya el modelo de datos. Para esto, en “command shell”, en la ubicación del proyecto general deberás de poner:

python manage.py makemigrations

Luego deberás de poner:

python manage.py migrate

Con estos dos comandos, crearás el modelo de datos en la base de datos “db.sqlite3” que se creó cuando creaste el proyecto.

Si quieres ver la base de datos te debes primero de bajar un visualizador de sqlite. Una opción es entrar a: <http://sqlitebrowser.org/>y bajarte el visualizador que se encuentra en la página.

**CARGAR INFORMACIÓN**

Al crear el modelo, estás creando la base de datos con los campos, pero todavía no tienes informacion en esta base de datos.

Para esto del repositorio de github <https://github.com/carlosgl87/desarrollo_soluciones_web/tree/master/django_tutorial/assets>

te deberás de bajar y copiar a tu proyecto la carpeta “management” y el archivo “pet\_data.csv”.

El archivo “pet\_data.csv” contiene un archivo con información de las mascotas. Esta contiene los mismos campos que la base de datos que hemos creado. La carpeta contiene un script que te permitirá copiar la información a la base de datos.

La carpeta pegala dentro de la carpeta adoptions y el archivo con la información crealo en el repositorio inicial.

Luego en “command shell” en la ubicación inicial del proyecto deberás de poner:

python manage.py load\_pet\_data

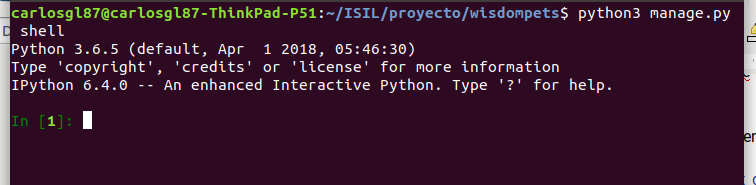
Esto ejecutará el programa de la carpeta “management” y copiará la información del archivo “.csv” a la base de datos en sqllite.

**INTERACTUAR CON LA BASE DE DATOS**

Una vez que tienes la base de datos con datos de mascotas puedes interactuar con esta base de datos.

Para esto dento de la carpeta inicial deberás de poner:

python manage.py shell



Al hacer esto dentro del shell que estes usando podrás realizar consultas a la base de datos.

Para esto, primero deberás de importar el modelo de mascotas. Para esto primero coloca:

from adoptions.models import Pet

Con esto podrás acceder a la información que se encuentra en la base de datos. Por ejemplo, puedes “pedir” a la base de datos todos los objetos, para esto deberás de poner:

pets = Pet.objects.all()

Pets, en este caso será una lista de objetos. Como toda lista, para acceder al primer elemento de la lista deberás de poner la posición 0 dentro de corchetes

pet = pets[0]

La variable pet será un objeto que tiene la información de la primera mascota de la base de datos, para acceder a su nombre poner:

pet.name

Por otro lado, si se quiere sacar información de la mascota con id 1, se deberá de poner:

pet = Pet.objects.get(id=1)

CREAR UNA VISTA DE ADMINISTRADOR

Para crear una vista de administrador