



Entrenamiento de Python-Django - 3

Final

Resumen: Hoy vamos a aprender a usar AJAX y Websockets con Django.

Versión: 1.1

Contenido

I	Preámbulo	2
II	Reglas generales	3
III	Reglas específicas de hoy	4
IV	Ejercicio 00	5
V	Ejercicio 01	7
VI	Ejercicio 02	9
VII	Ejercicio 03	10
VIII	Ejercicio 04	11
IX	Presentación y evaluación por pares	12

Capítulo I

Preámbulo

Charlar

El gato (*Felis catus*) es un pequeño mamífero carnívoro. Es la única especie domesticada de la familia Felidae y a menudo se le denomina gato doméstico para distinguirlo de sus miembros salvajes. Puede ser un gato doméstico, un gato de granja o un gato asilvestrado; este último se mueve libremente y evita el contacto humano. Los humanos valoran a los gatos domésticos por su compañía y su capacidad para cazar roedores. Diversos registros felinos reconocen unas 60 razas de gatos.

Fuente.



Figura I.1: Gato chateando en chatroulette.



¿Por qué mencionamos a los gatos aquí? Te preguntarás. Bueno, todos estos... Los temas fueron escritos por escritores franceses, y en francés, a un gato se le llama "Chat". Hay un juego de palabras, ¿ves? ¡Maudits Français!

Capítulo II

Reglas generales

- Su proyecto debe realizarse en una máquina virtual.
- Su máquina virtual debe tener todo el software necesario para completar su proyecto. Estos software deben configurarse e instalarse.
- Puede elegir el sistema operativo que utilizará para su máquina virtual.
- Debe poder utilizar su máquina virtual desde una computadora del clúster.
- Debe utilizar una carpeta compartida entre su máquina virtual y su máquina host.
- Durante tus evaluaciones utilizarás esta carpeta para compartir con tu repositorio.

Sus funciones no deben cerrarse inesperadamente (fallo de segmentación, error de bus, doble liberación, etc.), salvo por comportamientos indefinidos. Si esto ocurre, su proyecto se considerará no funcional y recibirá un 0 en la evaluación.

Te animamos a crear programas de prueba para tu proyecto, aunque este trabajo no tenga que entregarse ni se calificará. Esto te permitirá evaluar fácilmente tu trabajo y el de tus compañeros. Estas pruebas te resultarán especialmente útiles durante la defensa. De hecho, durante la defensa, puedes usar tus propias pruebas o las del compañero que estás evaluando.

Envía tu trabajo al repositorio Git asignado. Solo se calificará el trabajo en el repositorio Git. Si Deepthought se encarga de calificar tu trabajo, lo hará después de la evaluación por pares. Si se produce un error en alguna sección de tu trabajo durante la calificación de Deepthought, la evaluación se detendrá.

Capítulo III

Reglas específicas de hoy


- La única biblioteca de Javascript que puedes usar es JQuery.

Su entrega consistirá en un proyecto único de Django. No se dividirá en ejercicios, como es habitual. Cada uno de ellos aporta una funcionalidad al proyecto. Esta funcionalidad y su implementación se evaluarán en esta ocasión.

- Debes salir de la aplicación de administración predeterminada.
- Junto con su proyecto, debe entregar un archivo requirements.txt (con el comando 'pip freeze') que enumera las bibliotecas que su proyecto necesitará ejecutar.

Capítulo IV

Ejercicio 00

	Ejercicio 00
Ejercicio 00: ¡AJAX mi fórmula!	
Directorio de entrega: ex00/	
Archivos a entregar:	
Funciones permitidas: Ninguna	

Cree un nuevo proyecto llamado d09. En este proyecto, cree una aplicación llamada cuenta.

El objetivo de este ejercicio es diseñar un sistema de conexión/desconexión que se comunique únicamente gracias a AJAX.

En esta aplicación, implementará la URL 127.0.0.1:8000/account que debe conducir a una página que puede tener dos comportamientos diferentes según el contexto:

- El usuario no está conectado: la página debe mostrar un formulario de conexión estándar (usuario y contraseña). La comunicación con el servidor solo debe usar AJAX y debe ser de tipo POST para conectarse.

Si el formulario no es válido, los errores deben aparecer en la página.

Si la forma es válida, debe desaparecer y adoptar un nuevo comportamiento.

Esto, por supuesto, sin actualizar la página.

- El usuario ya está conectado: la página debe mostrar el siguiente texto: "Iniciado sesión como <usuario>", siendo <usuario> reemplazado por el nombre utilizado por el usuario para conectarse así como el botón Cerrar sesión que permite la desconexión.

Este botón debe comunicarse con el servidor mediante AJAX y el método 'POST'.

Una vez cerrada la sesión, el texto y el botón no deben mostrarse más en la página y esta debe adoptar el otro comportamiento.

La página nunca debe haber sido actualizada.

Si la página se actualiza 'manualmente', debe volver al comportamiento que tenía antes de actualizarse. actualización (que no incluye la visualización del error).


Puedes utilizar bootstrap.



Formulario de autenticación, ¡gratis!

Capítulo V

Ejercicio 01

	Ejercicio 01
Ejercicio 01: Chat básico	
Directorio de entrega: ex01/	
Archivos a entregar:	
Funciones permitidas: Ninguna	

Crea una aplicación llamada 'chat'.

En esta aplicación, debes crear una página con tres enlaces. Cada uno debe dirigir a una sala de chat diferente.

Los nombres de estas habitaciones deben estar en la base de datos. Debe crear un modelo adecuado.


Cada uno de estos enlaces debe conducir a otra página que contenga un chat funcional estándar. Cada chat debe tener la siguiente especificación:

- Debe utilizar 'jquery' como única biblioteca frontend, así como los Websockets para comunicarse con el servidor (no AJAX).
- Sólo está disponible para usuarios conectados.
- El nombre del chat debe aparecer en algún lugar.
- Varios usuarios deben poder conectarse (por si acaso...).
- Un usuario puede publicar un mensaje (lo habías adivinado, ¿verdad?).
- Un mensaje enviado por un usuario debe ser visible para todos los usuarios que se han unido a la sala de chat (todos saben lo que es una sala de chat, ¿verdad? ¿No has leído ese preámbulo?).
- Los mensajes deben aparecer en la parte inferior y mostrarse en orden ascendente (este es para ti, junto al calentador, ¿verdad?), junto con el nombre del usuario que los publicó.

- Los mensajes no deben desaparecer. Un mensaje no debe reemplazar a uno anterior. El orden de los mensajes no debe cambiar.
- Cuando un usuario se une a la sala de chat, el mensaje '<nombre de usuario> se ha unido al chat' debe aparecer para que lo vean todos los usuarios, incluido el que acaba de unirse. <nombre de usuario> se reemplaza por el nombre de dicho usuario, por supuesto.

Capítulo VI

Ejercicio 02

	Ejercicio 02
Ejercicio 02: Historia	
Directorio de entrega: ex02/	
Archivos a entregar:	
Funciones permitidas: Ninguna	


En este ejercicio mejorarás tu chat agregándole un historial de mensajes.

Cuando un nuevo usuario se une a la sala de chat, debe ver los últimos tres mensajes que se han publicado en esta sala de chat, de arriba a abajo, del más antiguo al más nuevo.

Una vez más, solo puedes usar JQuery como bibliotecas frontend y Websockets para comunicarte con el servidor.

Capítulo VII

Ejercicio 03

	Ejercicio 03
Ejercicio 03: Lista de usuarios	
Directorio de entrega: ex03/	
Archivos a entregar:	
Funciones permitidas: Ninguna	

En este ejercicio, mejorarás tu chat añadiendo una lista de usuarios conectados. Esta vez, se actualizará automáticamente.

Cuando el usuario se une a la sala de chat, debe poder acceder a la lista de conectados. usuarios (y debe aparecer en la lista).

Esta lista de usuarios debe estar claramente separada de la lista de mensajes (otro `<div>` u otro contenedor html).

Cuando un usuario se une a una sala de chat, su nombre debe aparecer en la lista, junto con otros usuarios conectados.

Cuando un usuario abandona la sala de chat, su nombre debe desaparecer de la lista de usuarios conectados y debe aparecer el mensaje "<nombre de usuario> ha abandonado el chat" después de los mensajes publicados. (<nombre de usuario> será el nombre del usuario que acaba de abandonar el chat).


Una vez más, solo puedes usar JQuery como bibliotecas frontend y Websockets para comunicarte con el servidor.



Antes que nada, intenta construir una lógica funcional. No buscamos optimización... todavía.

Capítulo VIII

Ejercicio 04

	Ejercicio 04
Ejercicio 04: Desplazamiento	
Directorio de entrega: ex04/	
Archivos a entregar:	
Funciones permitidas: Ninguna	

Haz que tu chat sea presentable configurando tu lista de mensajes en un contenedor de tamaño fijo. Si el número de mensajes excede el contenedor, deben desaparecer en la parte superior y debe aparecer una barra de desplazamiento lateral.

Además, la barra de desplazamiento debe aparecer siempre con el cursor hacia abajo para que los últimos mensajes siempre se muestren primero.

Capítulo IX

Presentación y evaluación por pares

Entrega tu tarea en tu repositorio de Git como de costumbre. Solo se evaluará el trabajo dentro de tus repositorios durante la defensa. No dudes en verificar los nombres de tus carpetas y archivos para asegurarte de que sean correctos.



El proceso de evaluación se realizará en el computador del evaluado.
grupo.