D09 - Django-Python Training

Django - Ajax - Websockets

Damien Cojan dcojan@student.42.fr
42 Staff pedago@staff.42.fr

Summary: Today we'll find out how to use AJAX and Websockets with Django.

Contents

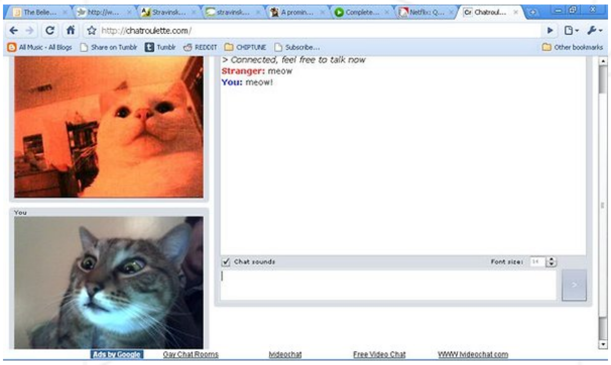
Chapter I

Preamble

Cat

The domestic cat (*Felis silvestris catus)* Following is the subspecies of the domestication of the wild cat, carnivorous mammal of the cat family. It is one of the leading pet and now has fifty races different re known by the bodies certification. In many countries, the cat is part of legislation on domestic carnivores like dogs and ferrets.

Source.



> Connected, feel free to talk now
Stranger: meow
You: meow!

figure I.1 - Chat chatting on chatroulette.

No exercise during the day speaks of this cat there. I prefer to clarify before a doubt sets in. Do not think that I doubt your intelligence right? I prefer to just anticipate

. . . for those in the bottom near the radiator ... ... it's always those in the background anyway. * Sighs *

Chapter II

Instructions

- Only this page serve as a reference: Do not firm the hallway noise.
- The subject can change up to an hour before rendering.
- If no contrary information is explicitly present, you should assume that the versions of languages and frameworks used are (or later):
  - Python 3
  - HTML5
  - CSS 3
  - Javascript EM6
  - Django 1.9
  - psycopg2 2.6
- Unless otherwise indicated in the subject, the files in python each year on Python one (d01, d02 and d03) must have their end block if __name__ == '__main__': in order to insert the entry point in the case of a program, or tests in the case of a module.
- Unless otherwise indicated in the subject, each year days on Django will have its own application in the project to make for reasons peda- gogiques.
- The exercises are precisely ordered from simple to more complex. In no event shall we bear no attention or take into account a complex exercise if a simpler exercise is not very successful.
- Attention to the rights of your files and your directories.
- You must follow the rendering process for all your exercise: only the present work on your GIT repository will be evaluated in defense.
- Your exercises will be evaluated by your pool mates.
- You should leave in your repertoire no other file than those explicitly specified by the forward drills.
- Unless otherwise specified in the topic you should not include in your rendering:

- The files __ pycache__.
- Any migration.
  Warning, you are still advised to make the file migration / __ init__.py,
  it is not necessary, but simplifies the construction of migration. Do not add this file will not invalidate your rendering but you *must* be able to manage migration for correction in this case.
- The file created by the command collectstatic of manage.py (with the way the value of the variable STATIC_ROOT).
- The file Python bytecode (files with extension. pyc).
- database files (especially with sqlite).
- Any file or folder must or can be created by the normal behavior of the rendering work.

It is recommended that you modify your. gitignore in order to avoid accidents.

- When you need to get an accurate output in your programs, it is of course forbidden to a ffi expensive output precalculated instead of performing the exercise cor- rectly.
- You have a question ? Ask your neighbor to the right. Otherwise, try your left neighbor.
- Your reference manual called Google / man / Internet / ....
- Think discuss on the forum of your pool Intra!
- Read the examples carefully. They might require things that are not otherwise specified in the subject ...
- Please, by Thor and Odin! Refléchissez dammit!

chapter III

Specific Rules of the day

- The only javascript library that you have to use is JQuery
- Your rendering will take the form of a single project Django. He will not have the usual cutting exercises. Each rajoutent project functionality. It is this and its implementation that will be noted.
- You must leave the default management application.
- You must make your project a file requirement.txt (via 'pip freeze') listing the libraries needed to run your project.

Chapter IV

Exercise 00

	Exercise: 00 00 Exercise: Ajax
my formulah!	
Render Folder: *ex* 00 / Files to make: The necessary files for your application	
Permitted functions: See setpoint	
Remarks: n / A	

Create a new project named d09 and in this project an application named account.

The objective of this exercise is to design a system connection / disconnection communicating only through AJAX.

You must implement this application in the url 127.0.0.1:8000/account which must return a page that may have two behaviors diff erent in two cases of FIG:

- The user is not connected: The page has a ffi expensive standard login form (login, password), except that communication with the server to validate the form should only use AJAX and must be of type POST.

  If the form is not valid, or the errors must be a ffi Chees on the page. If the form is valid, it must disappear

  and adopt another behavior. All this without the page is refreshed ... at any time.

- The user is already connected: The page has a ffi expensive the phrase " Logged as <user> " <User> is replaced by the name with which the user is connected, and a button Logout to log out.

This button should communicate with the server via AJAX and the method 'POST'. Once logged out, the sentence and the button must vanish from the page and it should adopt another behavior. All this without the page is refreshed at any time ... always

In case the page is refreshed manually ', this must regain the compor- ment where you left off (this does not include a ffi errors chage).

You can use Bootstrap.

AuthenticationForm is present!

Chapter V

Exercise 01

	Activity: 01 year 01: Basic
Chat	
Render Folder: *ex* 01 / files to make: The necessary files for your application	
Permitted functions: See setpoint	
Remarks: n / A	

Create a new application named 'cat'.

In this application you must create a page ffi singing three links that should lead to Chacun 'Chatroom' different.

The name of these rooms must be in the database, you will need to create a suitable model.

Each of these links should lead to another page containing a standard functional cat. Each cat must have the following characteristics:

- He must use 'JQuery' as single library and the frontend Websockets to communicate with the server. (No AJAX)
- It is only accessible to registered users.
- The cat's name must appear somewhere.
- Many users need to connect to it (*since once you guessed . . .*) .
- It is possible for a user to post a message (*but you also know right?).*
- A message sent by a user must be visible to all other same chatroom (*everyone knows what a cat is not it? You read the preamble?).*

- Messages must ffi was expensive from top to bottom, from the oldest to newest (*this is for the original down to the bottom ... it's always those in the background anyway.)* accompanied by the name of the user who posted it.
- The messages should not disappear, a message should not replace another, the ordering of messages should not move.
- When a user logs, the message <Username> has joined the chat '
  should appear for all users including himself. <Username> is replaced by the name of the user.

Chapter VI

Exercise 02

	Exercise 02 Exercise 02:
History	
Render Folder: *ex* 02 / Files to make: The necessary files for your application	
Permitted functions: See setpoint	
Remarks: n / A	

In this exercise, you will improve your cat o ff him rant a history of mes- sages.

When a new user joins a chatroom, he must see ffi was expensive last three messages that have been posted this chatroom, from top to bottom, from the oldest to the newest.

Again only JQuery as frontend library and Websockets to communicate with the server are allowed.

Chapter VII

Exercise 03

	Exercise 03 Exercise
03: Userlist	
Render Folder: *ex* 03 / Files to make: The necessary files for your application	
Permitted functions: See setpoint	
Remarks: n / A	

In this exercise, you improve your cat o ff him rant a list of connected users that updates alone as much.

When the user connects to a chatroom, it must have access immediately to the list of connected users (including himself).

This user list must be separate visually the message list (another <div> or other container html).

When a user connects to a chatroom, its name should appear in the list of other connected users.

When a user leaves a chatroom, its name must be removed from the list of other users connected and the message <Username> has left the cat ' ap- should appear (<username> to replace the name of the user in question) in response to messages posted.

Again only JQuery as frontend library and Websockets to communicate with the server are allowed.

Look primarily to build a functional logic. The goal is not optimization

.

Chapter VIII

Exercise 04

	Exercise 04 Exercise
04: Scroll	
Render Folder: *ex* 04 / Files to make: The necessary files for your application	
Permitted functions: See setpoint	
Remarks: n / A	

Make your cat presentable by putting the list of messages in a container with a height and a fixed width. If the number of messages exceeds this height, the messages disappear too, and a scroll bar allows the scroll through.

In addition, the scroll bar should always be at the bottom, so that the latest messages are always in sight.