I/ HƯỚNG DẪN VÀ GIỚI THIỆU VỀ DJANGO

1. Django hoạt động như thế nào

Django tuân theo mẫu thiết kế MVT (Mẫu xem mô hình).

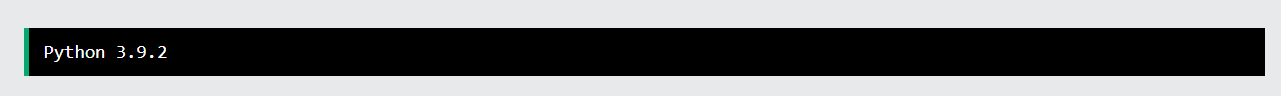
* Models - Dữ liệu ta muốn trình bày, thường là dữ liệu từ cơ sở dữ liệu.
* Xem - Trình xử lý yêu cầu trả về mẫu và nội dung có liên quan - dựa trên yêu cầu từ người dùng.
* Mẫu - Một tệp văn bản (như tệp HTML) có chứa bố cục của trang web, với logic về cách hiển thị dữ liệu

1. Các yêu cầu cần thiết

Django yêu cầu sử dụng python, để kiểm tra xem hệ thống của ta đã cài đặt Python chưa, hãy chạy lệnh này trong terminal:



Nếu python đã được cài đặt sẽ xuất hiện kết quả:



Để cài đặt Django, ta phải sử dụng trình quản lý gói như PIP, được bao gồm trong Python từ phiên bản 3.4.

Để kiểm tra xem hệ thống của ta đã cài đặt PIP chưa, hãy chạy lệnh này trong terminal:



Nếu pip đã được cài đặt sẽ xuất hiện kết quả:



1. Tạo môi trường ảo

Ta nên có một môi trường ảo dành riêng cho từng dự án Django và một cách để quản lý môi trường ảo là [“venv”](https://docs.python.org/3/tutorial/venv.html), được bao gồm trong Python.

Tên của môi trường ảo là sự lựa chọn của ta, trong hướng dẫn này, chúng ta sẽ gọi nó myworld.

Nhập các mục sau trong dấu nhắc lệnh, nhớ điều hướng đến nơi ta muốn tạo dự án của ta:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Điều này sẽ thiết lập một môi trường ảo và tạo một thư mục có tên "myworld" với các thư mục con và tập tin, như thế này:

Ảnh có chứa ảnh chụp màn hình, màu đen, Hình chữ nhật

Mô tả được tạo tự động

Sau đó, ta phải kích hoạt môi trường, bằng cách gõ lệnh này:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Khi môi trường được kích hoạt, ta sẽ thấy kết quả này trong dấu nhắc lệnh:

Ảnh có chứa ảnh chụp màn hình, văn bản, hàng

Mô tả được tạo tự động

Lưu ý: Ta phải kích hoạt môi trường ảo mỗi khi ta mở dấu nhắc lệnh để làm việc trên dự án của ta.

1. Cài đặt Django

Bây giờ, khi ta đã tạo ra một môi trường ảo, chúng ta đã sẵn sàng để cài đặt Django.

Django được cài đặt bằng pip, với lệnh này:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Ta có thể kiểm tra xem Django có được cài đặt hay không bằng cách hỏi số phiên bản của nó như thế này:



Nếu Django được cài đặt, ta sẽ nhận được kết quả với số phiên bản:



1. Tạo dự án

Khi ta đã đưa ra một tên phù hợp cho dự án Django của ta, như ở đây: my\_tennis\_club, điều hướng đến nơi trong tập tin hệ thống ta muốn lưu trữ mã ( trong môi trường ảo ), Giờ sẽ điều hướng đến myworld thư mục và chạy lệnh này trong terminal:



Django tạo ra một my\_tennis\_club thư mục trên máy tính của ta, với nội dung này:

Ảnh có chứa văn bản, Phần mềm đa phương tiện, phần mềm, ảnh chụp màn hình

Mô tả được tạo tự động

1. Chạy dự án

Bây giờ ta đã có một dự án Django, ta có thể chạy nó và xem nó trông như thế nào trong một trình duyệt.

Điều hướng đến /my\_tennis\_club thư mục và thực hiện lệnh này trong terminal:



Mở một cửa sổ trình duyệt mới và nhập [127.0.0.1:8000](http://127.0.0.1:8000/) trong thanh địa chỉ.

Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

1. Tạo ra App

App là một App web có ý nghĩa cụ thể trong dự án của ta, giống như một trang chủ, một hình thức liên lạc hoặc cơ sở dữ liệu thành viên

1. Cách tạo App

Ta sẽ đặt tên cho App của ta members.

Bắt đầu bằng cách điều hướng đến vị trí đã chọn nơi ta muốn lưu trữ App, trong trường hợp của tach là my\_tennis\_club thư mục và chạy lệnh dưới đây.

Máy chủ vẫn đang chạy và ta không thể viết lệnh, nhấn [CTRL] [BREAK] hoặc [CTRL] [C] để dừng máy chủ và ta sẽ quay lại ảo môi trường.



Django tạo một thư mục có tên members trong dự án của ta, với nội dung này:

Ảnh có chứa văn bản, Phần mềm đa phương tiện, phần mềm, ảnh chụp màn hình

Mô tả được tạo tự động

1. Views trong Django

Views “view”  là các hàm Python lấy các yêu cầu http và trả về phản hồi http, như tài liệu HTML.

Một trang web sử dụng Django có đầy đủ các Views với các tác vụ khác nhau và nhiệm vụ.

Views thường được đặt trong một tệp có tên views.py nằm trên thư mục App của ta

Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

Sau đó hãy thay thế nó như thế này để ta có thể thấy tác dụng của nó trong chủ đề tiếp theo

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

1. URLS trong Django

Với mỗi App chúng ta tạo, phải tạo them tập tin có tên urls.py

Các tập tin urls.py ta vừa tạo là cụ thể cho App members. Ta có để thực hiện một số định tuyến trong thư mục gốc my\_tennis\_club. Điều này có vẻ phức tạp, nhưng bây giờ, chỉ cần làm theo hướng dẫn phía dưới.

Có một tập tin được gọi là urls.py trên my\_tennis\_club thư mục, mở tệp đó và thêm include mô-đun trong import tuyên bố, và cũng thêm một path() chức năng trong urlpatterns[] danh sách, với các đối số sẽ định tuyến người dùng đi qua 127.0.0.1:8000/.

Sau đó, tập tin của ta sẽ trông như thế này

Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

Trong cửa sổ trình duyệt, nhập [127.0.0.1:8000/members/](http://127.0.0.1:8000/members/) trong thanh địa chỉ.

Ảnh có chứa ảnh chụp màn hình, văn bản, phần mềm, màn hình

Mô tả được tạo tự động

1. Tạo các trang mới trong Django

Dễ thấy rằng Django phải có html và nó sẽ được tạo trong 1 folder có tên là template bên trong thư mục “app”, ví dụ ở đây ta sẽ đặt nó là myfirst.html có nội dung như sau

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

Sau đó ta phải sửa đổi tệp tin view.py để phù hợp với trang html chúng ta tạo

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Mặt khác ta phải nói với Django rằng một App mới được tạo ra

Điều này được thực hiện trong settings.py tập tin trong thư mục my\_tennis\_club .

Tra cứu INSTALLED\_APPS[] liệt kê và thêm members App như thế này

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Mô tả được tạo tự động

Sau đó chạy lệnh này:



Rồi ta runserver lại và truy cập vào <http://127.0.0.1:8000/members/> sẽ được kết quả như sau

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

1. Models trong Django

Cho đến bây giờ trong hướng dẫn này, đầu ra là dữ liệu tĩnh từ các mẫu Python hoặc HTML.

Bây giờ chúng ta sẽ thấy Django cho phép chúng ta làm việc với dữ liệu như thế nào, mà không phải thay đổi hoặc tải lên các tập tin trong sự thịnh vượng.

Trong Django, dữ liệu được tạo trong các đối tượng, được gọi là Models “Models ” và thực sự là các bảng trong cơ sở dữ liệu.

1. Hướng dẫn tạo bảng

Để tạo một mô hình, điều hướng đến Models.py tập tin trong  thư mục /members/.

Mở nó và thêm một Member bảng bằng cách tạo một Member class, và mô tả các trường bảng trong đó:

Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Mô tả được tạo tự độngẢnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

1. Áp dụng bảng

Bây giờ khi ta đã mô tả một Models trong tệp Models .py, chúng ta phải chạy một lệnh để thực sự tạo bảng trong cơ sở dữ liệu.

Điều hướng đến thư mục /my\_tennis\_club/ và chạy lệnh này:



Django tạo một tệp mô tả các thay đổi và lưu trữ tệp trong /migrations/ thư mục:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Lưu ý rằng Django chèn một id trường cho bảng của ta, đó là một auto increment number (bản ghi đầu tiên nhận được giá trị 1, bản ghi thứ hai 2, v.v.), đây là mặc định hành vi của Django, ta có thể ghi đè lên nó bằng cách mô tả của riêng ta vào trường id.

Bảng chưa được tạo, ta sẽ phải chạy thêm một lệnh nữa, sau đó Django sẽ tạo và thực thi một câu lệnh SQL, dựa trên nội dung mới tập tin trong /migrations/ thư mục.

Chạy lệnh di chuyển:



* + Ta đã có 1 member trong cơ sở dữ liệu

1. Chèn dữ liệu trong Django

Bảng Thành viên được tạo trong mục trước trống rỗng.

Ta sẽ sử dụng trình thông dịch Python để thêm một số các thành viên của nó.

Để mở shell Python, nhập lệnh này:



Ở phía dưới, sau ba >>> viết như sau:



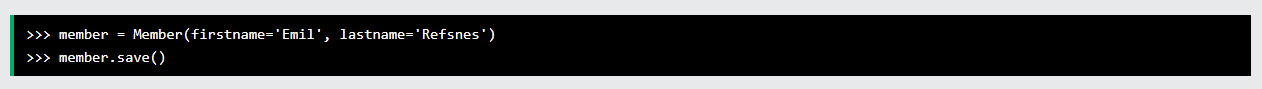
Nhấn [enter] và viết cái này để xem bảng Thành viên trống:



Điều này sẽ cung cấp cho ta một đối tượng QuerySet trống, như thế này:



Để thêm một bản ghi vào bảng, bằng cách thực hiện hai dòng này:



Thực hiện lệnh này để xem bảng Thành viên có thành viên không:

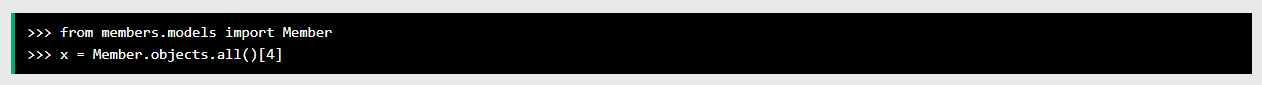


Hy vọng, kết quả sẽ như thế này:



1. Cập nhật dữ liệu đã thêm trong Django

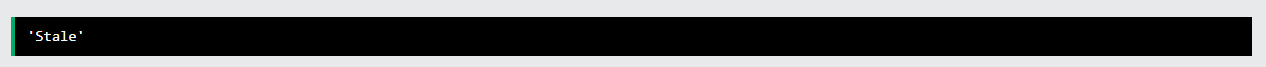
Để cập nhật các bản ghi đã có trong cơ sở dữ liệu, trước tiên chúng ta phải nhận bản ghi mà chúng ta muốn cập nhật:



X bây giờ sẽ đại diện cho thành viên ở chỉ số 4, đó là "Stale Refsnes", nhưng để chắc chắn, hãy cho chúng ta xem điều đó có đúng không:



Điều này sẽ cung cấp cho ta kết quả này



Bây giờ chúng ta có thể thay đổi các giá trị của bản ghi này:



1. Xoá dữ liệu trong Django

Để xóa bản ghi trong bảng, hãy bắt đầu bằng cách lấy bản ghi ta muốn xóa:



X bây giờ sẽ đại diện cho thành viên ở chỉ số 5, đó là "Jane Doe", nhưng để chắc chắn, chúng ta hãy xem điều đó có đúng không:



Điều này sẽ cung cấp cho ta kết quả này:



Bây giờ chúng ta có thể xoá dữ liệu:



Kết quả sẽ là(Điều này cho chúng ta biết có bao nhiêu mục đã bị xóa và từ đó Model.)



1. Cập nhật dữ liệu trong Django

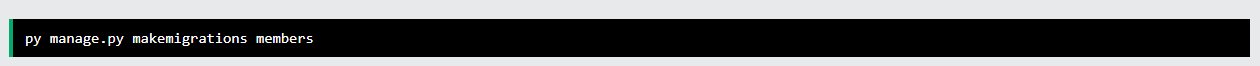
Để thêm một trường vào một bảng sau khi nó được tạo, hãy mở Models .py tập tin và thực hiện các thay đổi của ta:

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

Như ta thấy, chúng ta muốn thêm phone và joined\_date theo Models thành viên của chúng ta.

Đây là một thay đổi trong cấu trúc của Models và do đó chúng ta phải thực hiện di chuyển để nói với Django rằng nó phải cập nhật cơ sở dữ liệu:



Mà trong trường hợp này, sẽ dẫn đến một dấu nhắc, bởi vì ta cố gắng thêm các trường không được phép là null, đến một bảng đã chứa các bản ghi.

Như ta có thể thấy, Django hỏi ta có muốn cung cấp các trường cụ thể không giá trị, hoặc nếu ta muốn dừng di chuyển và sửa nó trong mô hình:

Ảnh có chứa văn bản, phần mềm, Phông chữ, Phần mềm đa phương tiện

Mô tả được tạo tự động

Ta sẽ chọn tùy chọn 2 và mở Models .py nộp lại và cho phép các giá trị NULL cho hai trường mới, rồi thực hiện lại. Khi đó ta đã có trường dữ liệu mới trong bảng

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

III/ Admin

1. Django Admin

Admin là một công cụ thực sự tuyệt vời trong Django, nó là giao diện người dùng của tất cả các Models của ta

Nó là miễn phí và sẵn sàng để sử dụng với Django:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

Để vào giao diện người dùng quản trị, hãy khởi động máy chủ bằng cách điều hướng đến thư mục /myworld và thực hiện lệnh này:



Trong cửa sổ trình duyệt, nhập [127.0.0.1:8000/admin/](http://127.0.0.1:8000/admin/) trong thanh địa chỉ.

Kết quả sẽ như thế này:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động

Lý do tại sao URL này đi đến trang đăng nhập quản trị viên Django có thể được tìm thấy trong urls.py tập tin dự án của ta:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Các urlpatterns[] danh sách yêu cầu đi để admin/ và gửi chúng đến admin.site.urls, đó là một phần của một App tích hợp đi kèm với Django và chứa rất nhiều chức năng và giao diện người dùng, một trong số đó là giao diện người dùng đăng nhập.

2. Tạo Admin

Để có thể đăng nhập vào App quản trị viên, chúng ta cần tạo một người dùng.

Điều này được thực hiện bằng cách gõ lệnh này trong Views lệnh:



Sau đó ta nhập tên đăng nhập, mật khẩu và các thông tin liên quan:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

Ở đây thấy rằng mật khẩu của ta không đáp ứng các tiêu chí, nhưng đây là môi trường thử nghiệm và dù sao tta cũng chọn tạo người dùng, bằng cách nhập y

Nếu ta nhấn [Enter], ta nên tạo thành công người dùng



Vào lại trang admin đăng nhập và nếu thành công ta sẽ được giao diện như sau:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

3. Giao diện của trang Admin

Để bao gồm Models, Thành viên trong giao diện quản trị viên, ta phải nói với Django rằng Models này sẽ hiển thị trong giao diện quản trị viên.

Điều này được thực hiện trong một tập tin được gọi là admin.py, và nằm trong thư mục App của ta, trong trường hợp này là thư mục members.

Mở nó ra, và nó sẽ trông như thế này:

Ảnh có chứa văn bản, Phông chữ, hàng, ảnh chụp màn hình

Mô tả được tạo tự động

Chèn một vài dòng ở đây để hiển thị Models Thành viên trong quản trị viên trang:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Bây giờ quay lại trình duyệt và ta sẽ nhận được kết quả này:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

Nhấp chuột Members và xem năm bản ghi chúng ta đã chèn trước đó trong hướng dẫn này:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

4. Thay đổi cách hiển thị của trang Admin

Khi ta hiển thị Models dưới dạng danh sách, Django hiển thị mỗi bản ghi dưới dạng biểu diễn chuỗi của đối tượng bản ghi, trong trường hợp của ta là "Member object (1)", " Member object (2)", v.v.:

Để thay đổi điều này thành một định dạng thân thiện với người đọc hơn, chúng ta có hai lựa chọn:

1. Thay đổi hàm biểu diễn chuỗi, \_\_str\_\_() của Models thành viên

Để thay đổi biểu diễn chuỗi, chúng ta phải xác định biểu diễn chuỗi \_\_str\_\_() chức năng của Models thành viên trong Models .py, như thế này:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Kết quả ta sẽ được như này:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

1. Đặt list\_details

Chúng ta có thể kiểm soát các trường để hiển thị bằng cách chỉ định chúng trong một list\_display tài sản trong admin.py tập tin.

Đầu tiên tạo ra một MemberAdmin() lớp và chỉ định list\_display tuple, như thế này:

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

Hãy nhớ thêm MemberAdmin dưới dạng argumet trong admin.site.register(Member, MemberAdmin).

Bây giờ quay lại trình duyệt và ta sẽ nhận được kết quả này:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

3. Cập nhật Member

Bây giờ chúng ta có thể tạo, cập nhật và xóa các thành viên trong cơ sở dữ liệu của chúng ta và chúng ta bắt đầu bằng cách cung cấp cho họ tất cả một ngày tham gia thành thành viên và cho họ thêm số điện thoại

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

Sau đó ấn vào Save ta sẽ được kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động

4. Thêm Members

Để thêm thành viên mới, nhấp vào nút "ADD MEMBERS" ở trên cùng bên phải góc:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

Ta sẽ nhận được một biểu mẫu trống nơi ta có thể điền vào các trường thành viên:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Biểu tượng máy tính

Mô tả được tạo tự động

Điền vào các trường và ấn Save

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

5. Xoá thành viên

Để xóa thành viên mới, ta có thể chọn thành viên và chọn hành động "Xóa thành viên đã chọn” ở mục Action

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

IV/ TRUY VẤN TRONG DJANGO

1. Giới thiệu

Truy vấn là một tập hợp dữ liệu từ cơ sở dữ liệu.

Truy vấn được xây dựng dưới dạng danh sách các đối tượng.

Truy vấn giúp dễ dàng lấy dữ liệu ta thực sự cần, bằng cách cho phép ta lọc và đặt hàng dữ liệu ở giai đoạn đầu.

Trong hướng dẫn này, ta sẽ truy vấn dữ liệu từ Member.

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Mô tả được tạo tự động

Trong views.py, ta có một quan điểm để thử nghiệm được gọi là testing nơi ta sẽ kiểm tra các truy vấn khác nhau.

Trong ví dụ dưới đây, ta sử dụng hàm .all() để có được tất cả các hồ sơ và lĩnh vực của Member mô hình:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Đối tượng được đặt trong một biến được gọi là mydata, và được gửi đến mẫu thông qua context đối tượng như mymembers, và trông như thế này:

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, đại số

Mô tả được tạo tự động

Như ta thấy, Models Member chứa 5 bản ghi và được liệt kê bên trong QuerySet là 5 đối tượng.

Trong ví dụ, ta có thể sử dụng mymembers đối tượng để tạo nội dung:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

Kết quả:

Ảnh có chứa văn bản, phần mềm, số, ảnh chụp màn hình

Mô tả được tạo tự động

2. Truy vấn lấy dữ liệu  
Có nhiều phương pháp khác nhau để truy vấn lấy dữ liệu

a. Phương thức giá trị

Các values() phương pháp cho phép ta trở lại mỗi đối tượng dưới dạng từ điển Python, với tên và giá trị là khóa / giá trị cặp:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Mô tả được tạo tự động

1. Trả lại các cột cụ thể

Các values\_list() phương pháp cho phép ta trở lại chỉ các cột mà ta chỉ định.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Kết quả:

Ảnh có chứa văn bản, phần mềm, Trang web, Website

Mô tả được tạo tự động

1. Trả lại các hàng cụ thể

Ta có thể lọc tìm kiếm để chỉ trả về các hàng / bản ghi cụ thể, bằng cách sử dụng filter()

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Mô tả được tạo tự động

Kết quả:

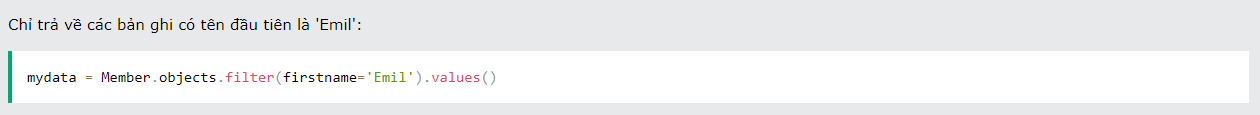
Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

3. Bộ lọc truy vấn

Các filter() phương thức được sử dụng để lọc tìm kiếm của ta và cho phép ta chỉ trả về các hàng khớp với thuật ngữ tìm kiếm.

Như chúng ta đã học trong chương trước, chúng ta có thể lọc các tên trường như thế này:

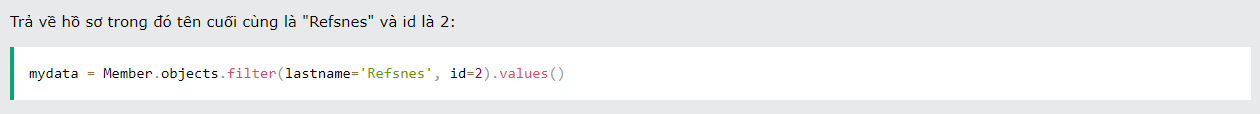


Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Các filter() phương thức lấy các đối số làm \*\*kwargs ( đối số từ khóa ), vì vậy ta có thể lọc trên nhiều trường bằng cách tách chúng bằng dấu phẩy



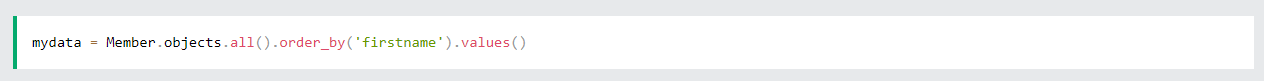
Kết quả:

Ảnh có chứa văn bản, phần mềm, Trang web, Phông chữ

Mô tả được tạo tự động

4. Truy xuất theo thứ tự

Để sắp xếp Truy vấn, Django sử dụng order\_by() phương pháp:



Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

…. Nhiều quá nói tụi nó vô W3 xem đi =))))