

Managers in django

Gonzalez Hernandez Cruz Omar

*Universidad Tecnológica de Tijuana.
TSU. Tecnologías de la información.
Entornos virtuales y negocios digitales.
Teacher: Parra Galaviz Ray Brunett*

Introduction

In this document, we will embark on an extensive exploration of the role of Django managers, a fundamental concept in the realm of web application development. Our aim is to provide you with a comprehensive understanding of managers, their functionalities, and their significance within the Django framework. We will delve into their definition, practical applications.

What are Managers

A manager in Django is a class that provides methods for performing queries and operations on the database in relation to a specific model. Each Django model has at least one default manager called objects, which allows you to interact with the database in a simple way. [1]

Apply this to your project

Django Managers



Creating a Custom Manager

To create a custom manager, you need to define a class that inherits from `models.Manager` and add custom methods to perform the operations you desire. Then, assign this class as an attribute in your model [1]

Custom managers

Custom managers in Django are managers that you define in your models to add specific functionality to database management. Although all models in Django have at least one default manager called "objects," you can create your custom managers to perform custom queries, apply specific filters, or carry out other database operations according to your application's needs. [1]

What are they used for [2]

Managers are used for the following tasks:

- Performing database queries to retrieve records that match certain criteria.
- Creating new records in the database.
- Updating existing records.
- Deleting records from the database.
- Performing aggregation and data analysis operations.

Features of managers in Django [2]

- Managers provide predefined methods such as `filter()`, `get()`, `create()`, `all()`, and more to interact with the database.
- You can define your custom managers in a model if you want to add specific methods for that model.
- Managers allow you to perform complex database queries through method chaining, making it easier to construct complex SQL queries without writing SQL directly.

Conclusion

In conclusion, Django managers are a fundamental component of web application development. They streamline database interactions, enhance code maintainability, and offer a wealth of functionalities to developers. Our exploration aims to equip you with the knowledge and skills to leverage managers effectively in your Django projects, enriching your capabilities as a web developer.

BIBLIOGRAFIAS IEEE

- [1] [En línea]. Available:
] <https://docs.djangoproject.com/en/4.2/topics/db/managers/>.
- [2] [En línea]. Available:
] <https://coffeebytes.dev/managers-o-manejadores-personalizados-en-django/>.