INHERITANCE IN DJANGO

Silva Sanchez Alvaro Daniel

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

Introduction

In this document we are going to see Within Django, model inheritance provides the possibility of generating new models using pre-existing base models, which involves the incorporation of their fields and functionalities. This notion is presented as a valuable resource for the reuse and expansion of existing models, offering exceptional versatility in different contexts.

Abstract base classes

Abstract base classes in Django are not intended to be stand-alone instances, but rather as foundations for other models, allowing fields and methods defined in them to be inherited by descendant models. To create an abstract base class, you set the "abstract" attribute to True within the Model Metaclass, which is a useful practice in organizing models and shared fields in Django applications.

Multiple table inheritance

Multiple table inheritance in Django is applied when you want to establish a model that works as a subclass of another model, and each of these models has its own table in the database. In this approach, fields from the parent model are incorporated into the table of the child model, and each model has the ability to include additional fields and methods, providing flexibility in structuring data and relationships.

Proxy models

Proxy models in Django are a way to create a new model that extends the functionality of an existing model. However, both models share the same database table and no new fields are added. Proxy models are often used to add custom methods to an existing model without modifying the original model.

Conclusion

In conclusion, model inheritance allows you to create new models based on existing models to reuse and expand code. abstract base classes serve as a foundation for other models, multi-table inheritance allows fields to be shared between models, and proxy models extend functionality without adding new fields, providing flexibility in developing django applications.