

Module 2: Models and Databases

Module 2 delves into Django's models and databases, crucial components for any web application. Models in Django are the single, definitive source for information about your data. They contain the essential fields and behaviors of the data you're storing. Django follows the DRY Principle, with the goal of defining your data model in one place and automatically deriving things from it. This module covers defining models, using Django's Object-Relational Mapper (ORM) to query the database, and the process of creating and applying migrations.

Defining Models

A model is a Python class that subclasses `django.db.models.Model`, where each attribute represents a database field. Django gives you an automatically-generated database-access API; creating, retrieving, updating, and deleting objects is done through the model class.

```
from django.db import models

class MyModel(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField()
```


Migrations

Migrations are Django's way of propagating changes you make to your models (adding a field, deleting a model, etc.) into your database schema. They are designed to be mostly automatic, but you'll need to know when to make migrations when to run them, and the common problems you might run into.

```
python manage.py makemigrations  
python manage.py migrate
```