

Django Notes - Overview and Installation

#overview

designed to make common web development tasks fast and easy

#django-model

it comes with an object-relational mapper in which you describe your database layout in Python code

Note:

ORM - lets to interact with databases in programming languages of different choice rather than forcing to use SQL.

makemigrations - looks at all your available models and creates migrations for whichever tables don't already exist

migrate - runs the migrations and creates tables in your database

django - free API to interact with models

code is not necessary

we can use django shell and import models to access it

#some example queries

#import model/table

```
>>> from news.models import Article, Reporter
      here, Articles and Reporter are tw tables/models in 'news' app
```

#retrieve all the object from table

```
>>> Reporter.objects.all()
```

insert new data to table using object

```
>>> r = Reporter(full_name='John Smith')
>>> r.save()
```

#can access with object and field name

```
>>> r.full_name
'John Smith'
```

#more option to read data

```
>>> Reporter.objects.get(id=1)
<Reporter: John Smith>
>>> Reporter.objects.get(full_name__startswith='John')
<Reporter: John Smith>
```

```
>>> Reporter.objects.get(full_name__contains='mith')
<Reporter: John Smith>
# Delete an object with delete().
>>> r.delete()
```

#django-admin interface

Django automatically creates an administrative interface

It allows authenticated users to add, change and delete objects.

To make use of this function we need to register a model in the admin site.

```
admin.site.register(models.Article)
```

django - design URLs

lets to design URLs

table of contents for app - contains a mapping between URL patterns and Python callback functions("views").

When a user requests a page, Django runs through each path, in order, and stops at the first one that matches the requested URL. (If none of them match, Django calls a special-case 404 view.)

django - views

responsible for doing one of two things - HttpResponseRedirect containing the content for the requested page, or raising an exception such as Http404

django - templates

Variables are surrounded by double-curly braces.

```
{{ article.headline }}
```

{{ article.pub_date|date:"F j, Y" }} uses a Unix-style "pipe" (the "|" character). This is called a template filter - to filter the value of a variable

Note:

template language doesn't provide exception handling, any exception raised from a template filter will be exposed as a server error

#installation

<https://docs.djangoproject.com/en/4.1/topics/install/>

#reference

<https://docs.djangoproject.com/en/4.1/intro/overview/>