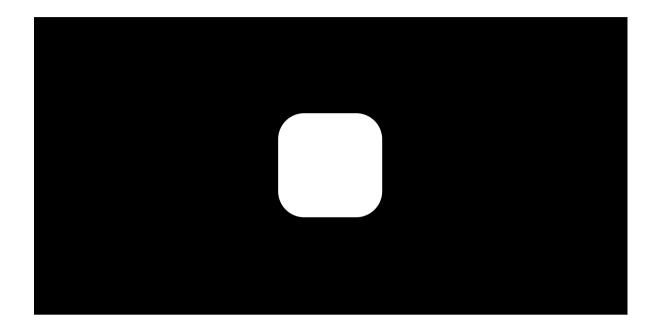
<u>Instructional Team</u> Office Hours <u>Progress</u> <u>Course</u> <u>Dates</u> **Discussion** (☆ Course / Lesson 5: Consuming APIs with NoSQL / Lesson 5 Content < Previous Next > **Part 1: Introduction** ☐ Bookmark this page

Part 1: Introduction

Mongo DB



Start of transcript. Skip to the end.

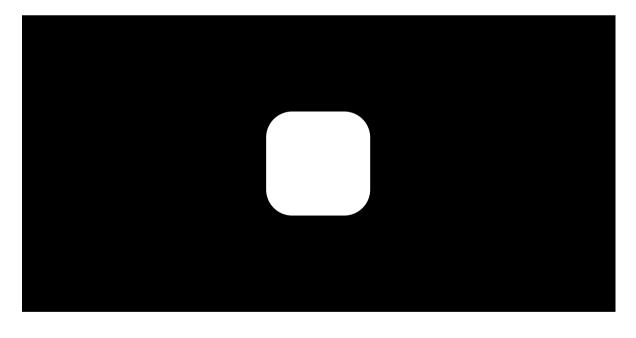
In this lesson, we're going to move on from looking at relational databases and we're going to look at NoSQL databases,

specifically one called MongoDB.

Peewee does a great job of making relational databases easy to use,

but MongoDB seems to take it to an extra level.

Introduction - an API perspective



Start of transcript. Skip to the end.

As with any language, when using Python to develop business applications,

you nearly always have to deal with additional runtime software

to help get your job done.

Examples of such software include message queue

in packages, such as Rabbit MQ, business rows engines, and of course

Video
<u>Download video file</u>

Transcripts

<u>Download SubRip (.srt) file</u>

<u>Download Text (.txt) file</u>

Code from the video

class BaseModel(Model):
 class Meta:
 database = database

-1--- D-----/D---M-4-1\.

```
crass berson(masewoder):
      This class defines Person, which maintains details of someone
      for whom we want to research career to date.
    person_name = CharField(primary_key = True, max_length = 30)
   lives_in_town = PCAW in the left (max_length = 40)
                                                        Next >
    nickname = CharField(max_length = 20, null = True)
class Job(BaseModel):
      This class defines Job, which maintains details of past Jobs
                                                                                          © All Rights Reserved
     held by a Person.
   job_name = CharField(primary_key = True, max_length = 30)
    start_date = DateField(formats = 'YYYY-MM-DD')
   end_date = DateField(formats = 'YYYY-MM-DD')
   salary = DecimalField(max_digits = 7, decimal_places = 2)
    person_employed = ForeignKeyField(Person, related_name='was_filled_by', null = False)
new_person = Person.create(
   person_name = 'Fred',
   lives_in_town = 'Seattle',
   nickname = 'Fearless')
   new_person.save()
aperson = Person.get(Person.person name == 'Fred')
```



© 2024 University of Washington | Seattle, WA. All rights reserved.

Help Center Contact Us Privacy Terms

Built on OPEN by RACCOONGANG

edX, Open edX and the edX and Open edX logos are trademarks or registered trademarks of edX Inc.