

Build CarModel and CarMake Django Models



Estimated time needed: 90 minutes

A dealership typically manages cars from one or more makes, and customers should be allowed to review the cars they purchased from a dealer.

In this lab, you will create the car model and car make related models in the Django app.

- A car model includes basic information such as its make, year, type, and dealer id.
- A car make includes basic information such as name and description.

Environment setup

If your Theia workspace has been reset and you want to continue from what you have done previously, `git clone` or pull from your created GitHub repository.

- Set up the Python runtime if Theia workspace has been reset.

1. 1

1. `python3 -m pip install -U -r requirements.txt`

Copied!

Note: You may need to perform models migrations for a new Theia environment.

To Build Car Model and Car Make models

You will need to create two new models in `djangoapp/models.py`:

- A `CarMake` model to save some data about a car make.
- A `CarModel` model to save some data about a car model.

You could find some hints in the code comments with the tag `<HINT>`.

- Create a car make Django model `class CarMake(models.Model)::`
 - Name
 - Description
 - Any other fields you would like to include in a car make
 - A `__str__` method to print a car make object
- Create a car model Django model `class CarModel(models.Model)::`
 - Many-To-One relationship to `CarMake` model (One car make has many car models, using a `ForeignKey` field)
 - Dealer Id (`IntegerField`) refers to a dealer created in Cloudant database
 - Name
 - Type (`CharField` with a choices argument to provide limited choices such as Sedan, SUV, and WAGON)
 - Year (`DateField`)
 - Any other fields you would like to include in a car model
 - A `__str__` method to print the car model object
- Run migrations for the models.

```
python3 manage.py makemigrations
```

and

```
python3 manage.py migrate
```

Note: If you see any errors related to model migrations, you could choose to delete the existing database `db.sqlite3` and re-run above migration again.

Refer to the previous Django ORM lab for more details:

[CRUD on Django Model Objects](#)

To register CarMake and CarModel models with the admin site

- First, you need to have a superuser for the admin site, if not created before.

Please use root as user name and root as password for your reviewer to login your app.

Take a screenshot for peer-review:

After you logged-in the admin site, please take a screenshot and name it as admin_login.jpg or admin_login.png for peer-review.

In addition, you may want to try log-out the admin user and you will be redirected to the admin login page again, and take a screenshot called admin_logout.jpg or admin_logout.png

Register CarMake and CarModel models

- You need to register the CarMake and CarModel in the admin site so you can conveniently manage their content (i.e., perform CRUD operations).
- You will also need to create CarModelInline so that you could manage CarModel and CarMake together on one page in the Admin site.

Note that for the dealer id field, you need to enter the integer dealer id you created in IBM Cloudant database. So that the car model can be associated with a dealer in the Cloudant database.

- After you register the new models, you can create a new car make with several cars for testing.

Take a screenshot for peer-review:

After you have filled the fields for a new car make on the admin page, please take a screenshot and name it as create_carmake.jpg or create_carmake.png for peer-review.

Also, after created the new car make, you should go back to the car make list and verify it was created successfully. Now take a screenshot of the current car make list with the newly added car make and name it updated_carmake_list.jpg or updated_carmake_list.png

Refer to the previous Admin site lab for more details:

[Django Admin Site](#)

Commit your updated project to GitHub

Commit all updates to the GitHub repository you created so that you can save your work.

If you need to refresh your memory on how to commit and push your code to GitHub in Theia lab environment, please refer to this lab [Working with git in the Theia lab environment](#)

Summary

In this lab, you have created the car model and car make related models in your Django app.

Author(s)

Yan Luo

Other Contributor(s)

Upkar Lidder

Priya

Changelog

Date	Version	Changed by	Change Description
2021-02-15	1.0	Yan Luo	Created new instructions for Capstone project
2022-09-20	1.1	K Sundararajan	Updated pip (packages installation) command
2022-10-04	1.2	K Sundararajan	Minor update made for clarity

© IBM Corporation 2023. All rights reserved.