Python Django Tutorial 2021

# Setup

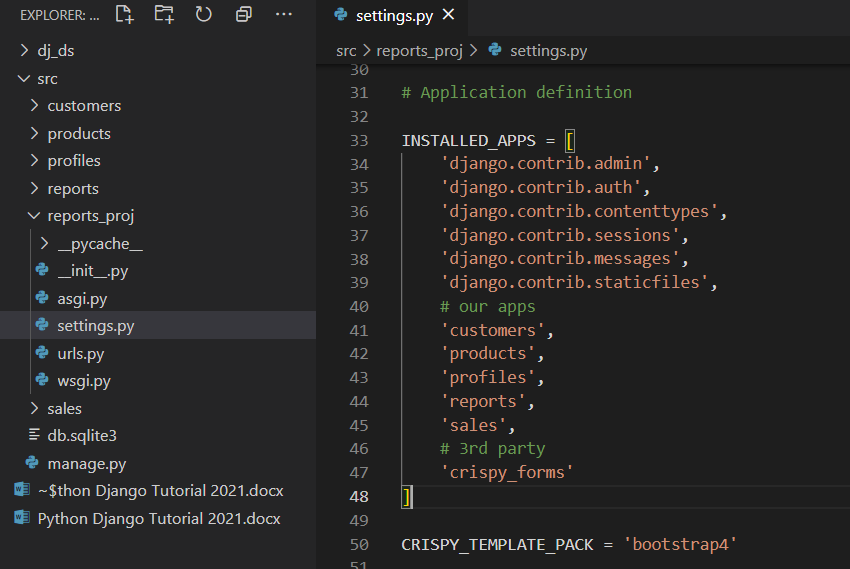
1. pip install virtualenv
2. virtualenv dj\_ds
3. source dj\_ds/bin/activate
4. pip install django
5. django-admin startproject reports\_proj
6. python manage.py migrate
7. python manage.py createsuperuser (dongpv/123456a@)
8. python manage.py startapp sales
9. python manage.py startapp reports
10. python manage.py startapp profiles
11. python manage.py startapp products
12. python manage.py startapp customers
13. python manage.py runserver

# Setup 2

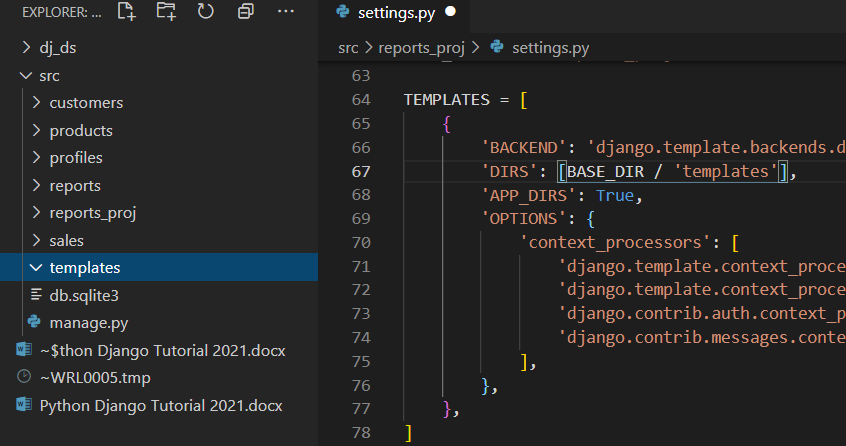
1. pip install pillow django-crispy-forms matplotlib seaborn pandas xhtml2pdf

# Setup 3

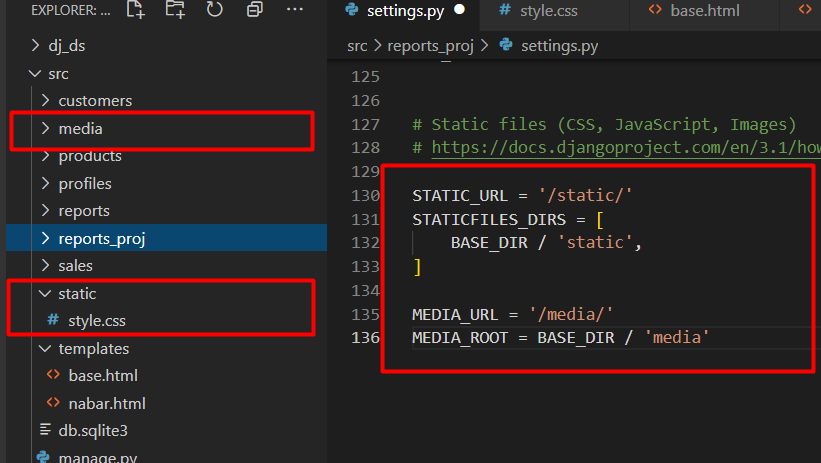
1. Setup apps



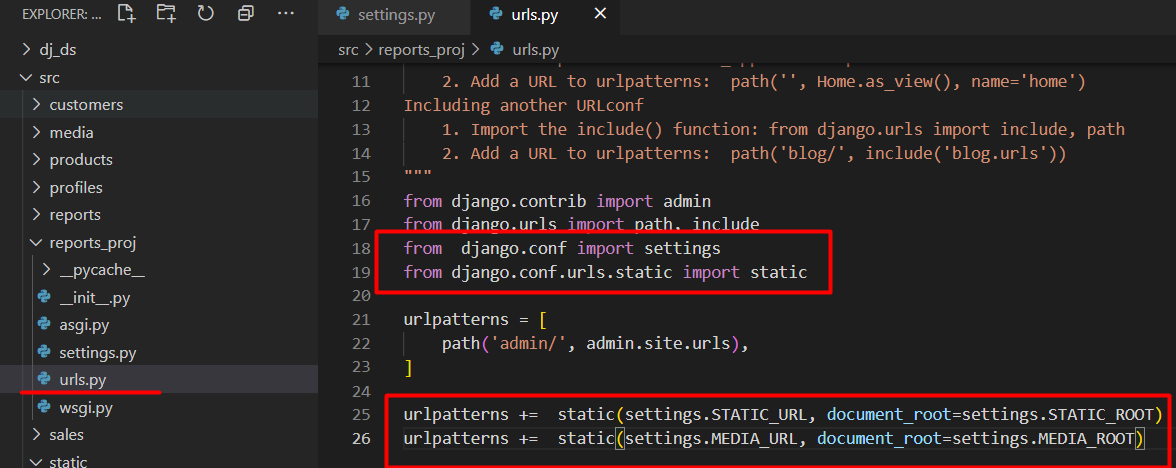
1. Add templates file



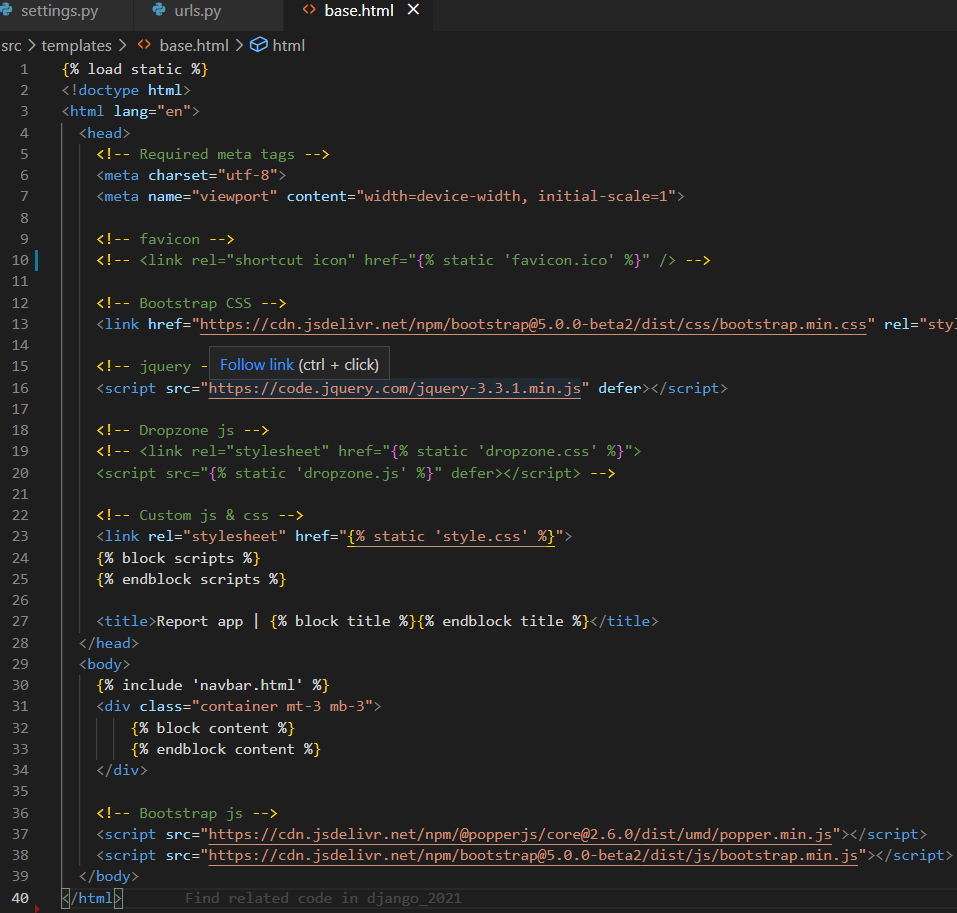
1. Add media and static files



1. Adding static\_url and media\_url to urls.py file

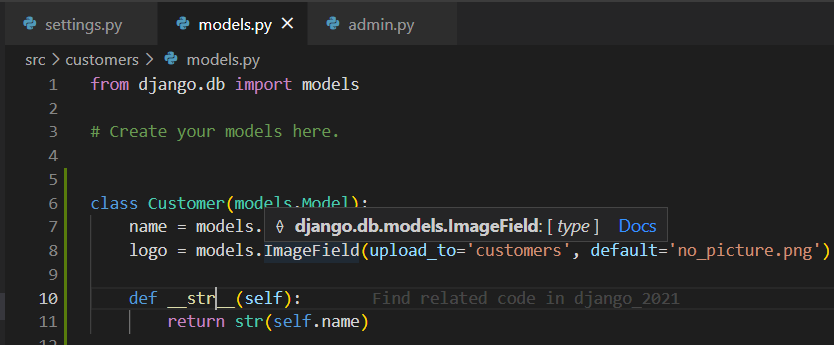


1. Add base.html code

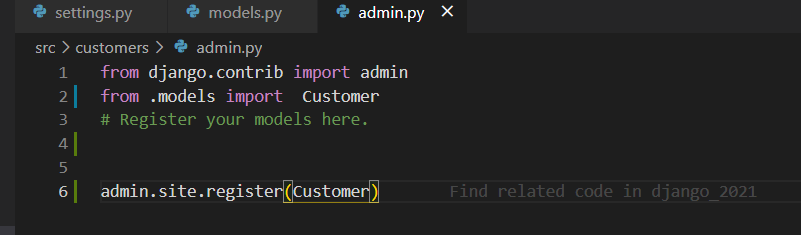


# Customer Model

1. Adding customer model



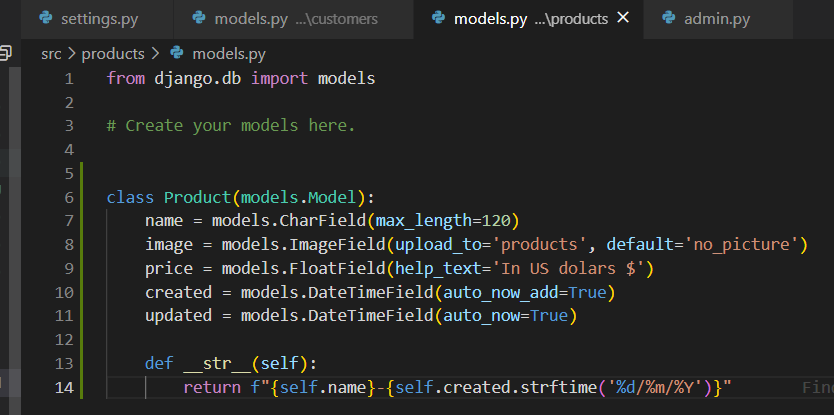
1. Register customer model



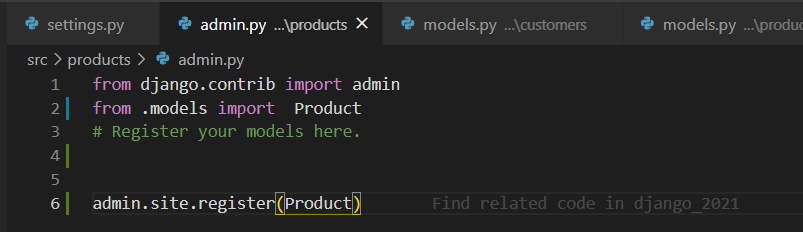
1. python manage.py makemigrations
2. python manage.py migrate

# Product Model

1. Adding product model



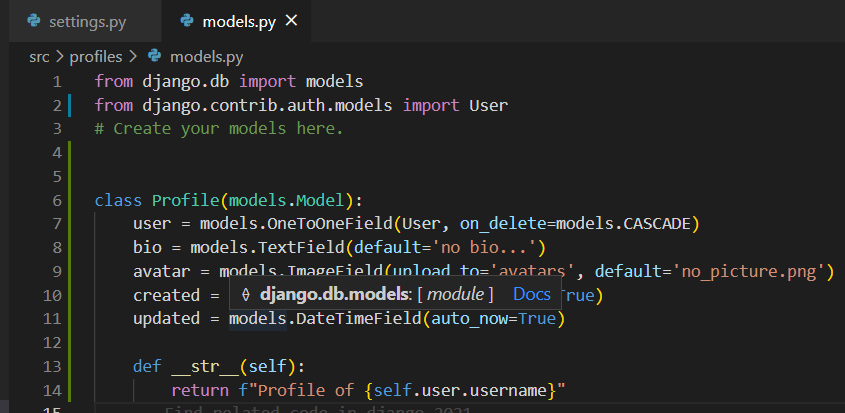
1. Register product model



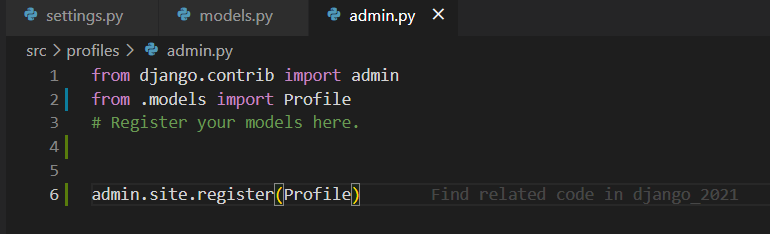
1. python manage.py makemigrations
2. python manage.py migrate

# Profile Model + POST\_SAVE Signal

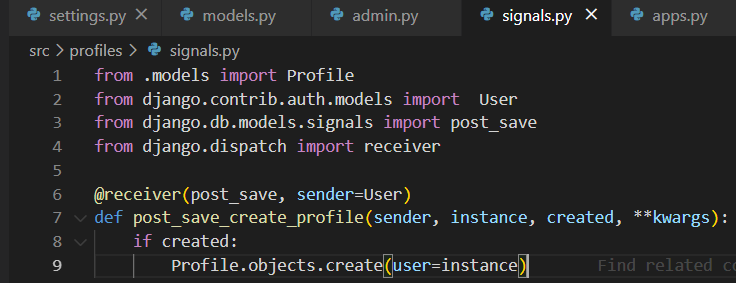
1. Adding profile model



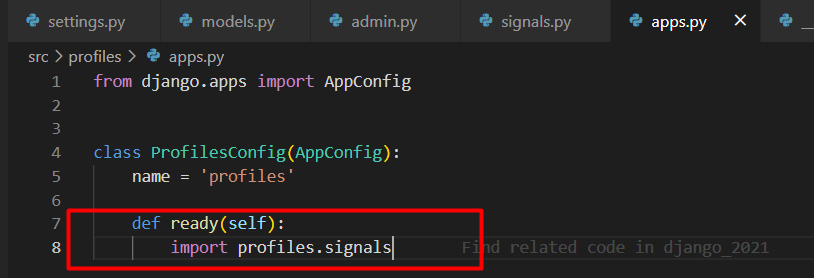
1. Register profile model



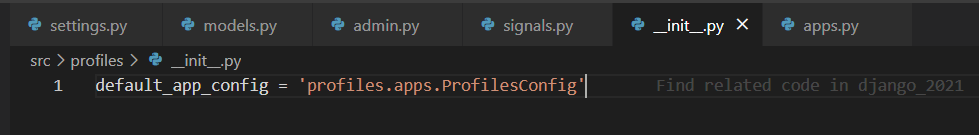
1. python manage.py makemigrations
2. python manage.py migrate
3. Add signals.py file



1. Import signals in apps.py



1. Add default\_app\_config to \_\_init\_\_py

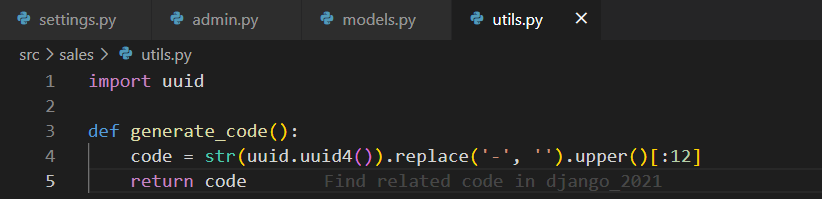


# Sale Model

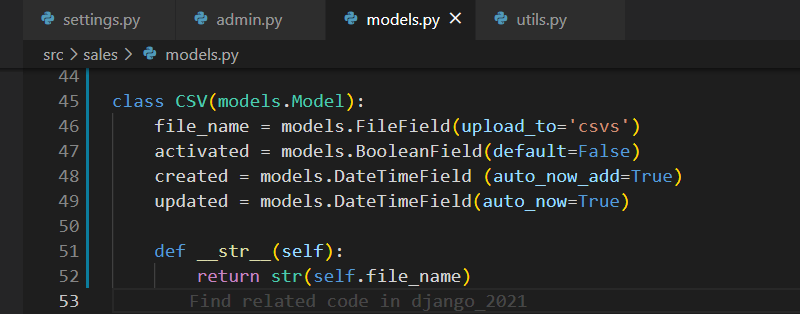
1. Adding positon, sale model



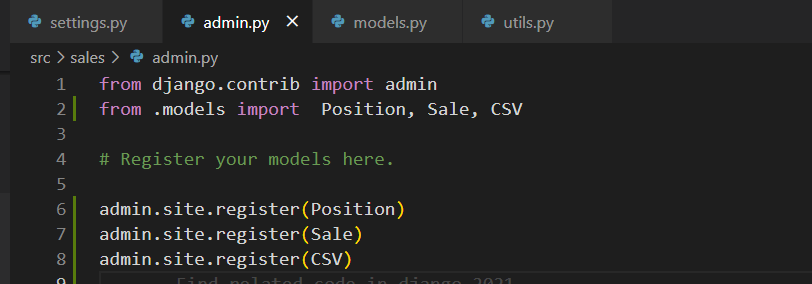
1. Add utils.py to generate transaction\_id in sale



1. Add csv model



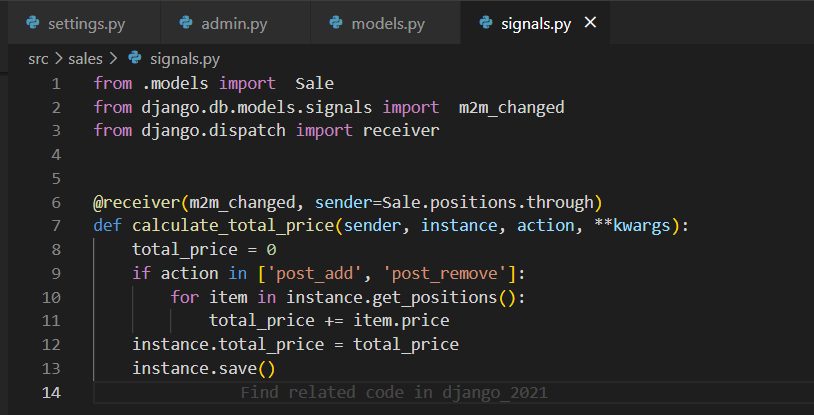
1. Register models



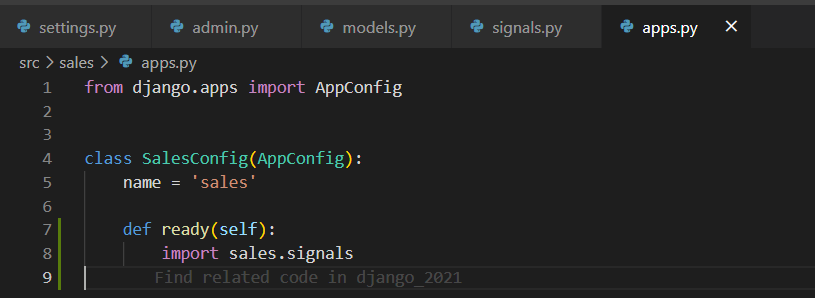
1. python manage.py makemigrations
2. python manage.py migrate

# m2m\_changed Signal

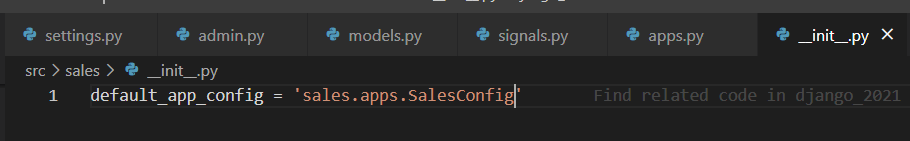
1. Add signals.py



1. Overwrite ready func

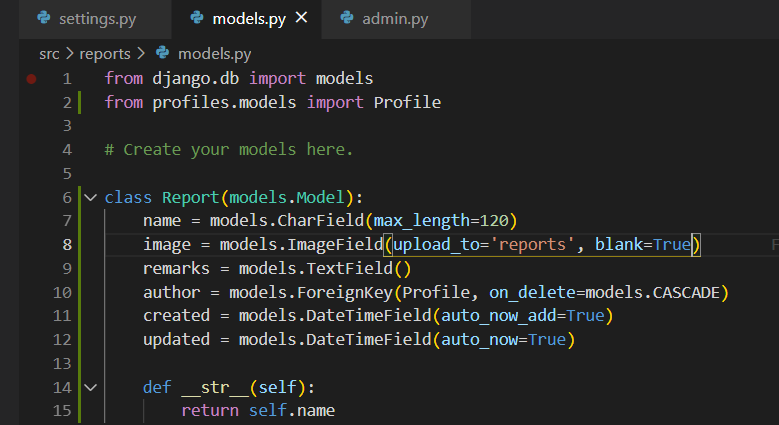


1. Add default\_app\_config

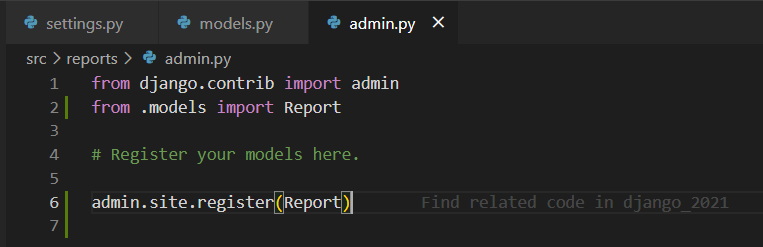


# Reports model

1. Add report model



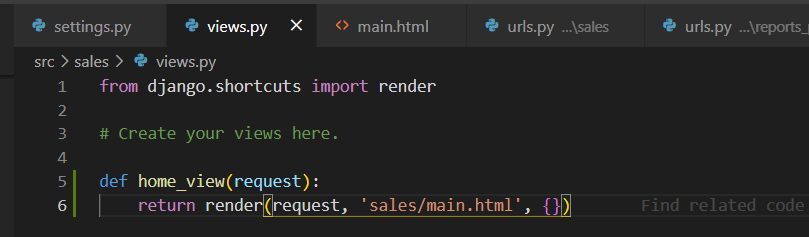
1. Register report model



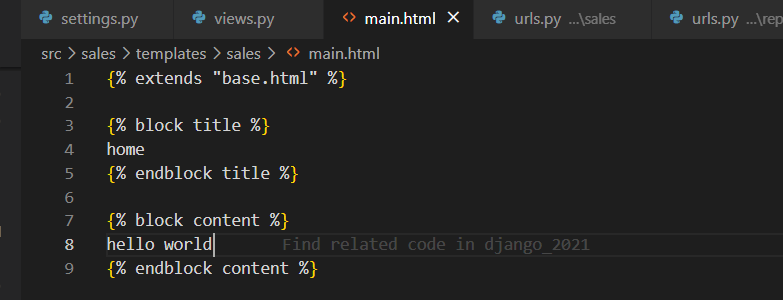
1. python manage.py makemigrations
2. python manage.py migrate

# First View and Template

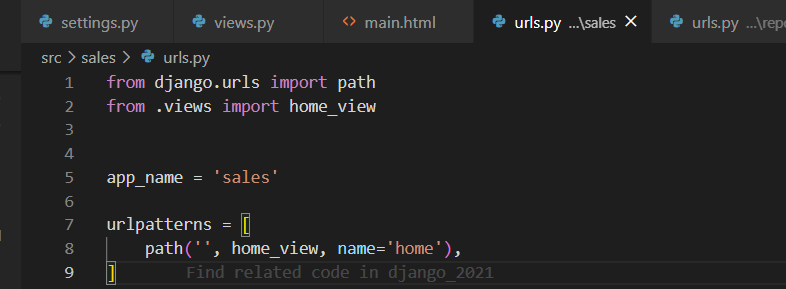
1. Add view to sales app



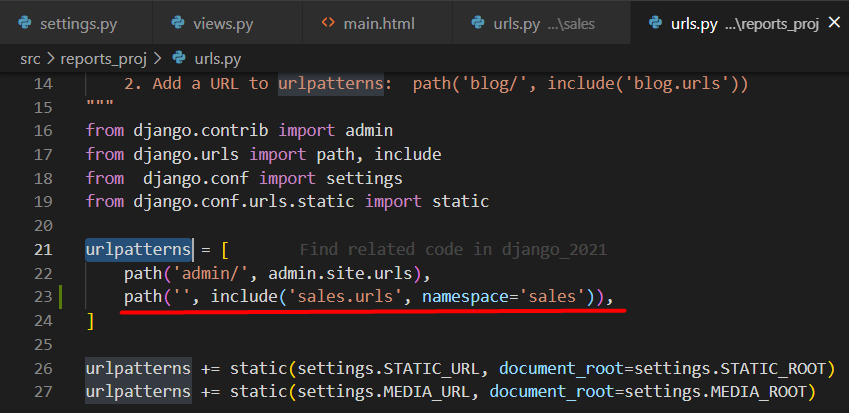
1. Add home.html template



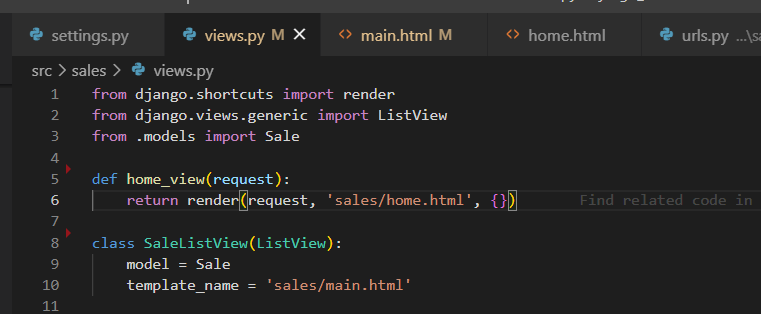
1. Add urls.py in sales app



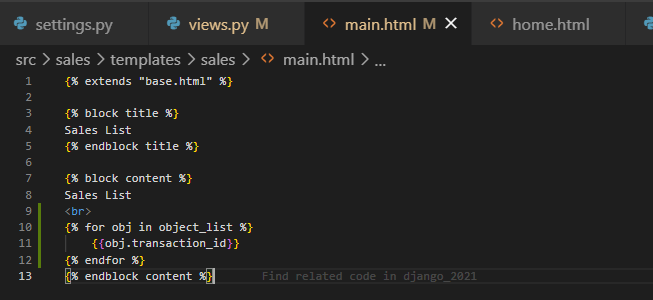
1. Include sales.urls im main urls.py



1. Add class ListView

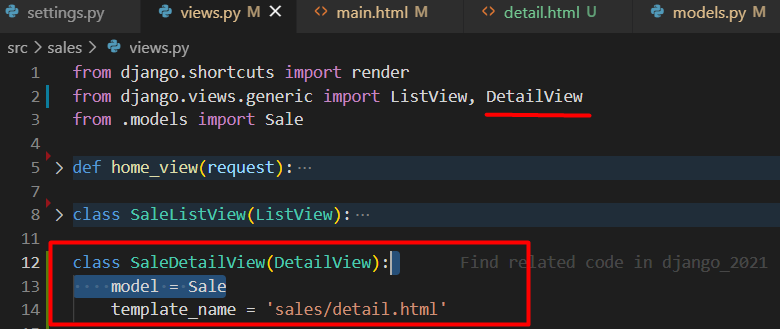


1. Add main.html

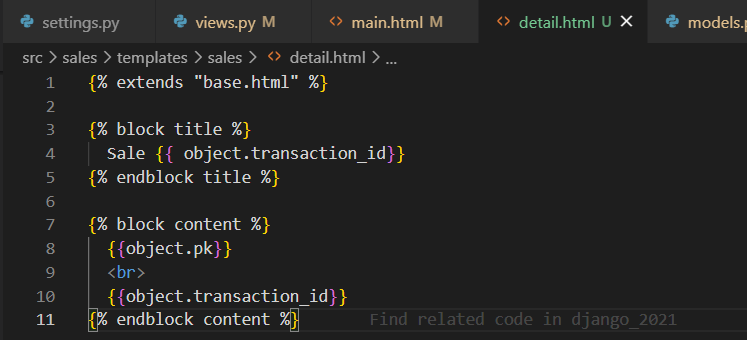


# Detail view

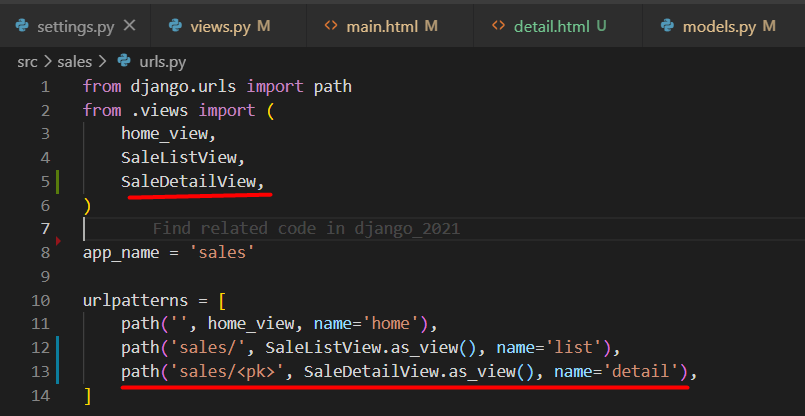
1. Add detailview



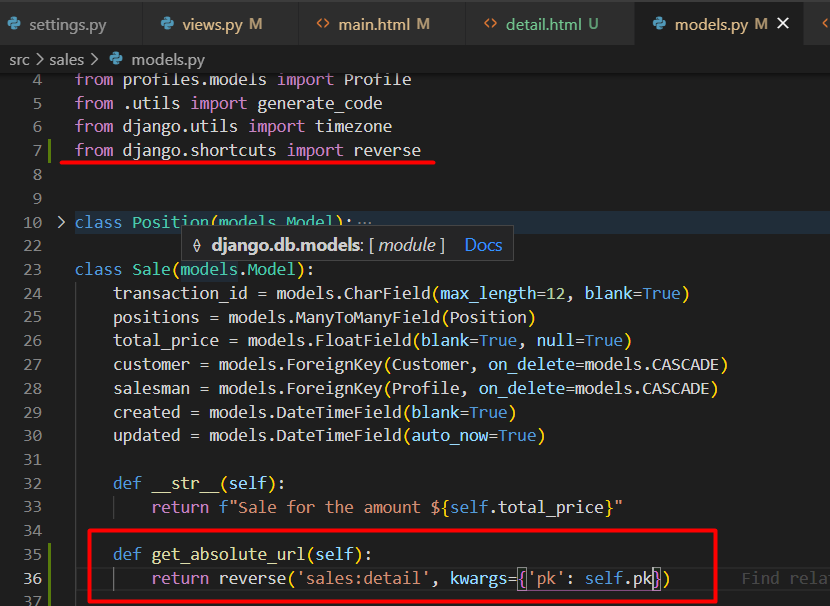
1. Add detail.html

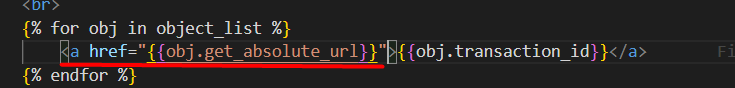


1. Add detail url



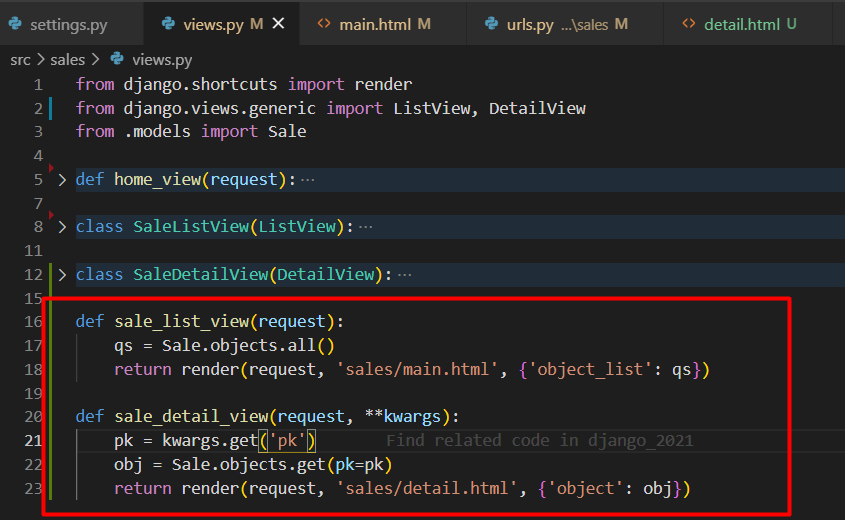
1. Add get\_absolute\_url func in sale model

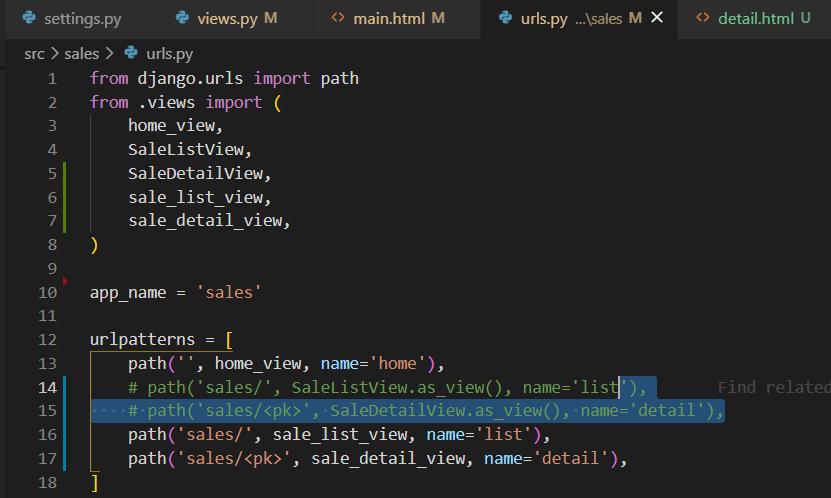




# Function base view

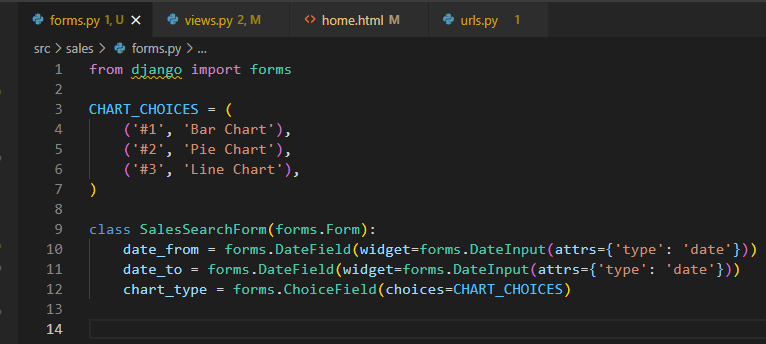
1. Sale list function base view and sale detail function base view



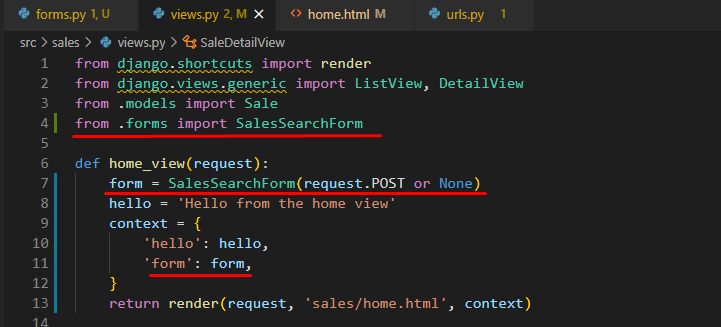


# Create search form

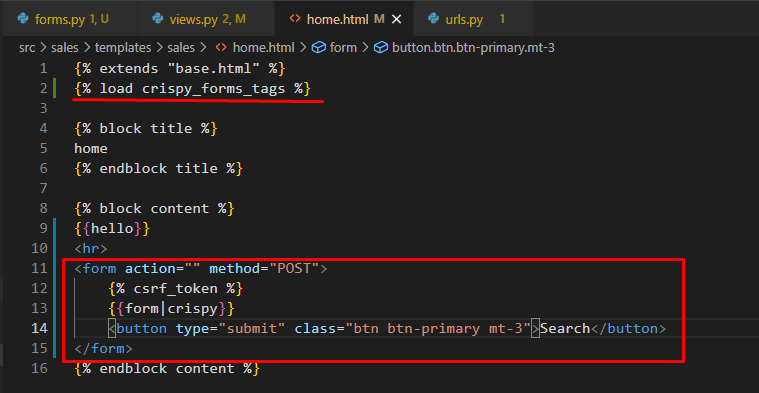
1. Add SalesSearchForm



1. Add SalesSearchForm to function base view

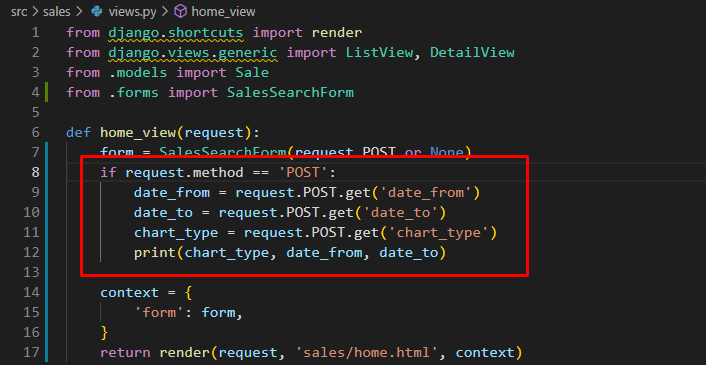


1. Add SalesSearchForm to home view



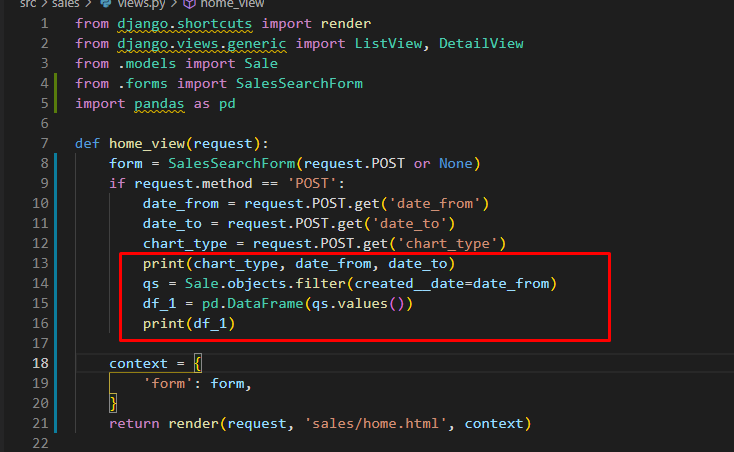
# Get data from search form

1. Get info from search form



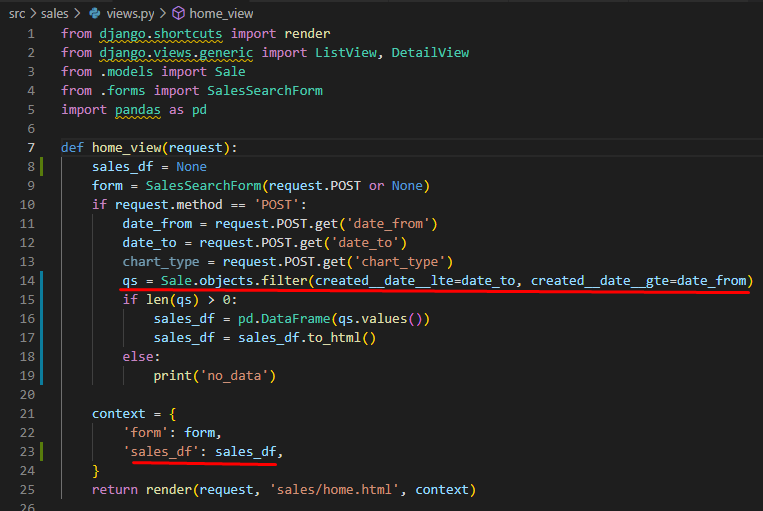
# First query and dataframes

1. Query set by date\_from

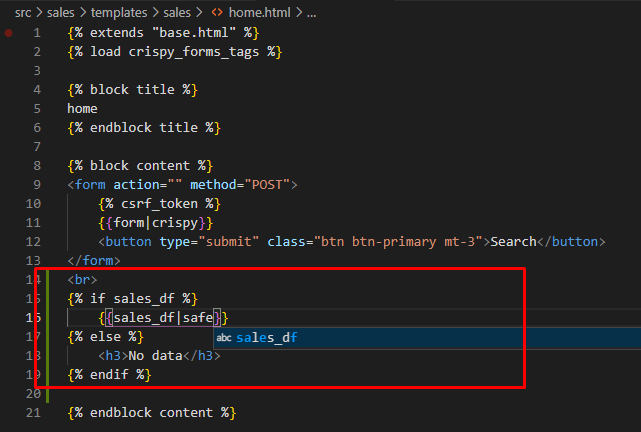


# Display dataframes in the template

1. Query set by date\_from and date\_to

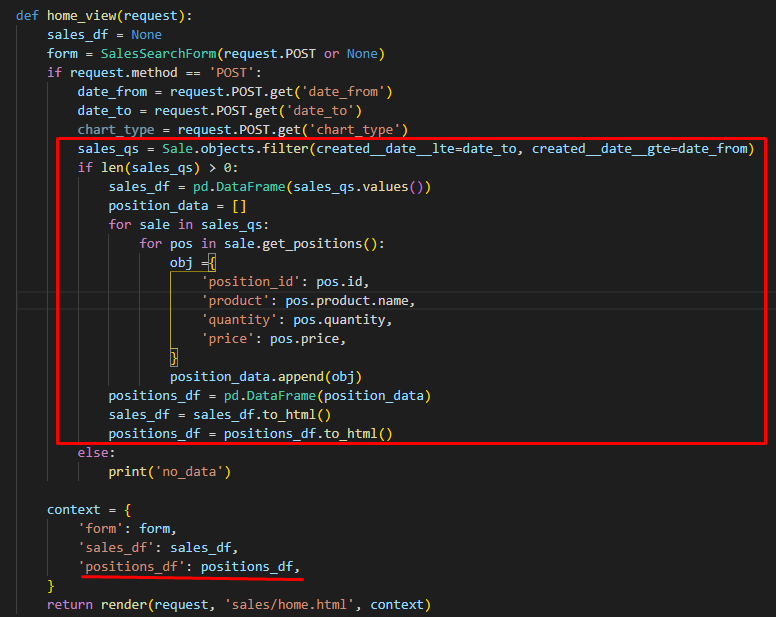


1. Display data frames in to view

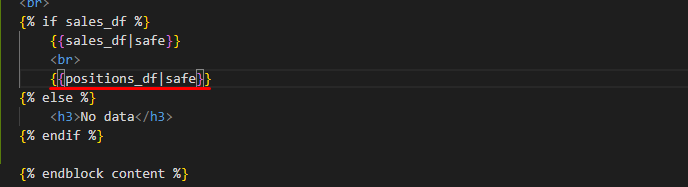


# Dataframe for the positions

1. Get positions data

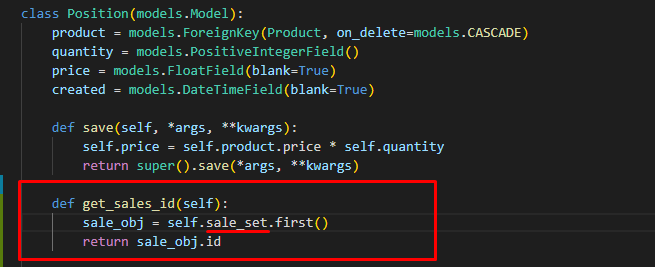


1. Display positions data in view

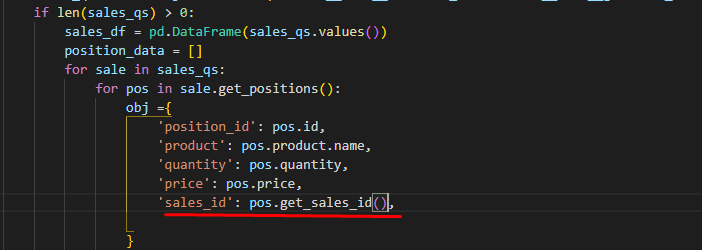


# Get sale id for position object

1. Get sale\_id

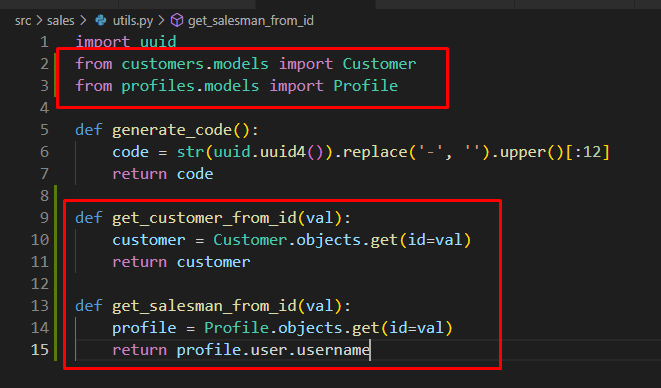


1. Get sale\_id in positions\_df

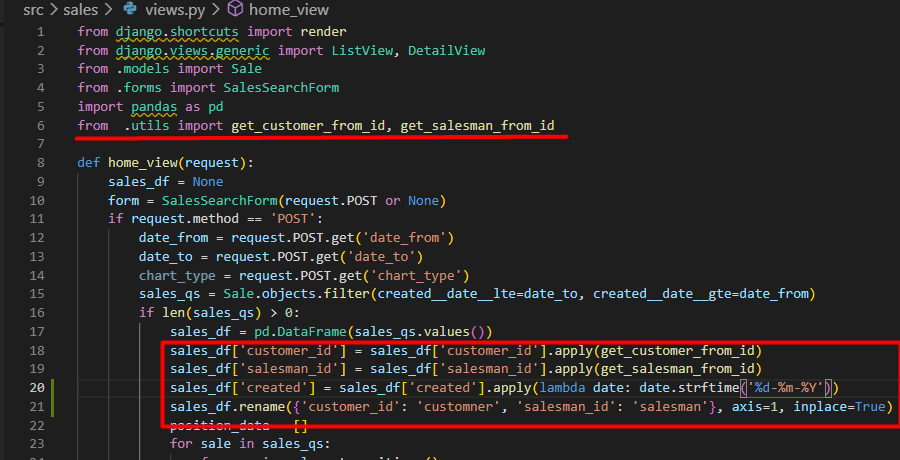


# The apply function

1. Add function get customer\_name and user\_name

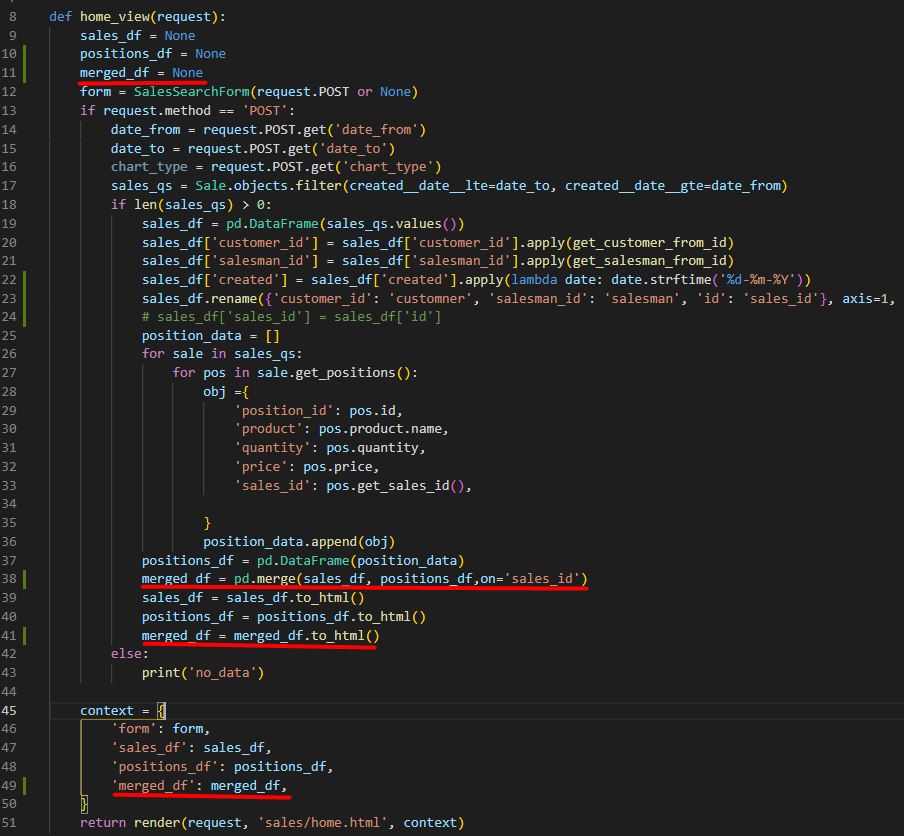


1. Displace name instead of id

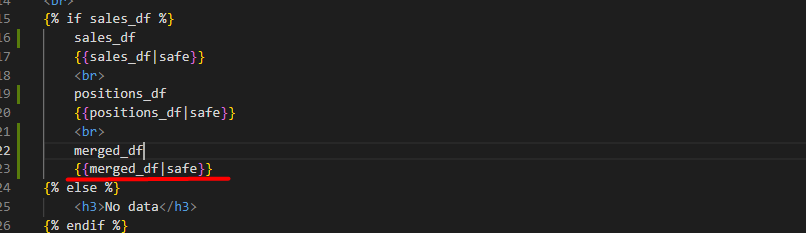


# Merge dataframes

1. Add merged dataframes

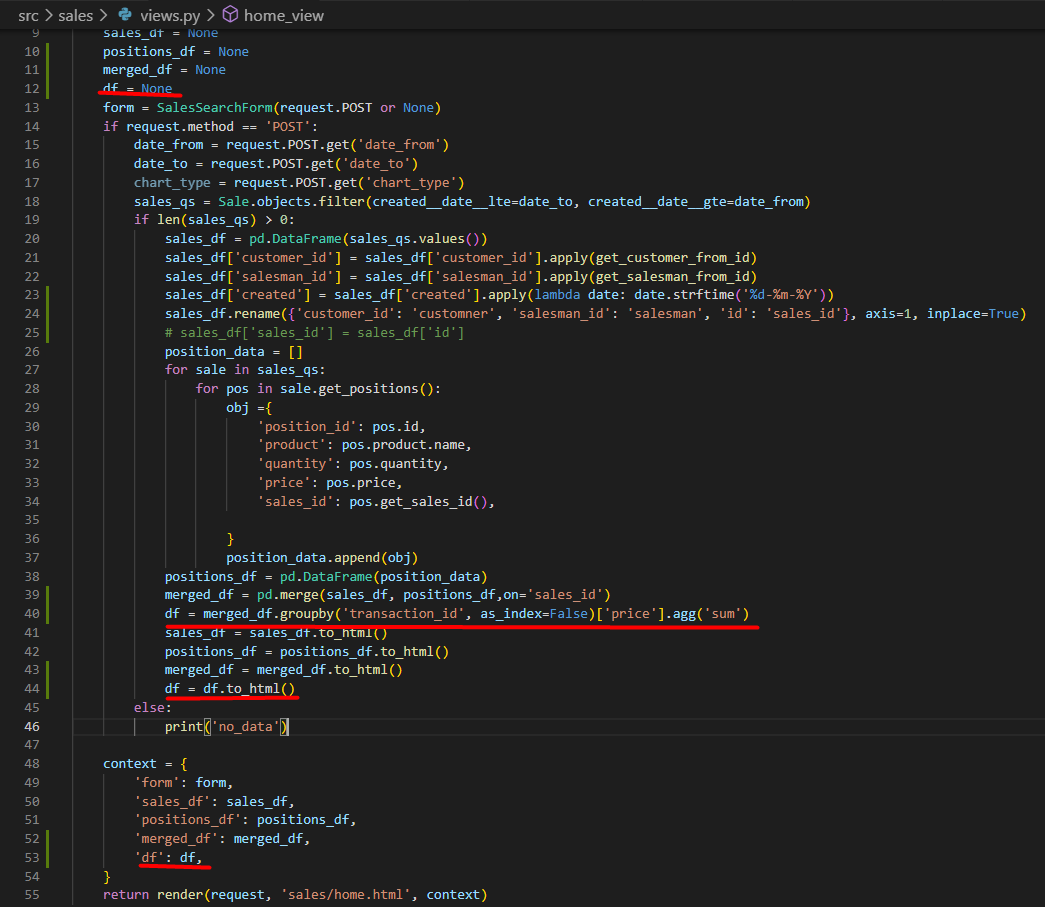


1. Display merged dataframes to view

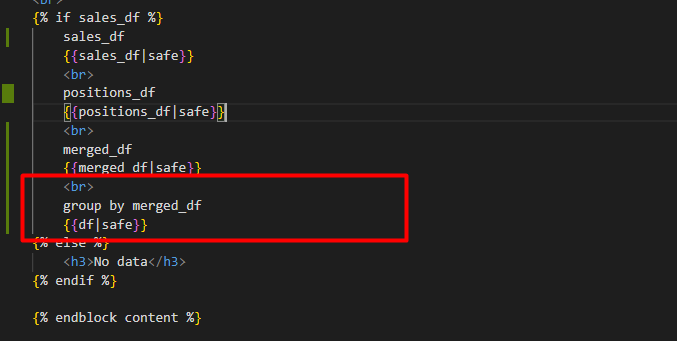


# Perform groupby

1. Add group by transaction\_id

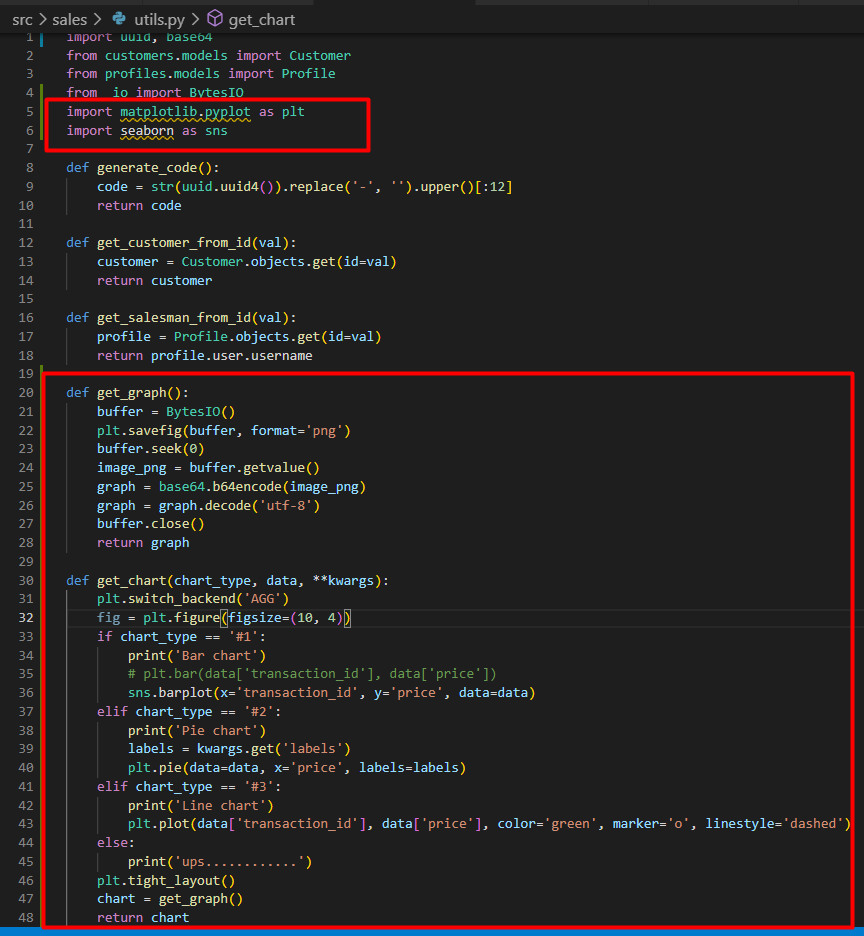


1. Show grouped data in view

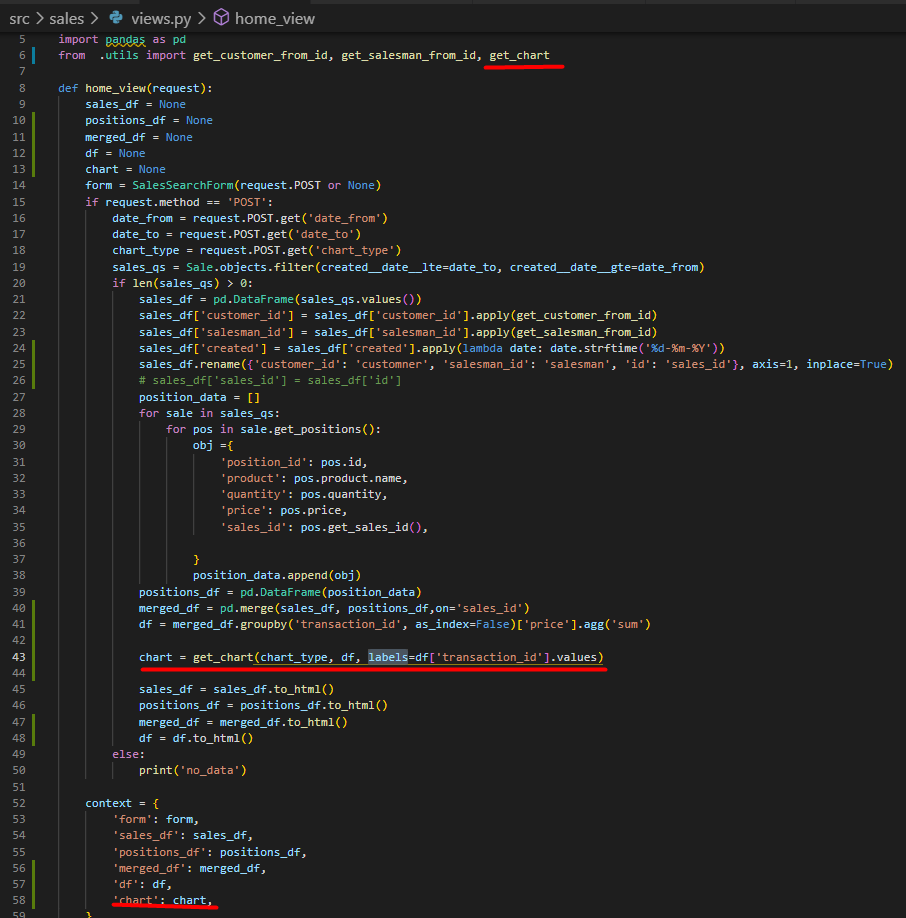


# Working on charts

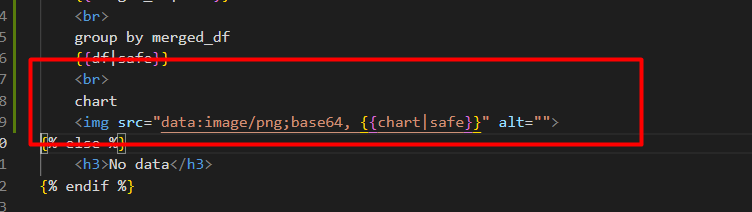
1. Prepare chart with matplotlib and seaborn



1. Get chart from home\_view function

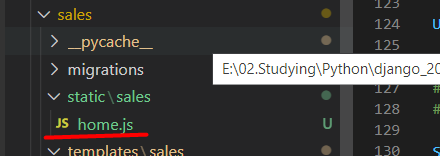


1. Displace chart in view

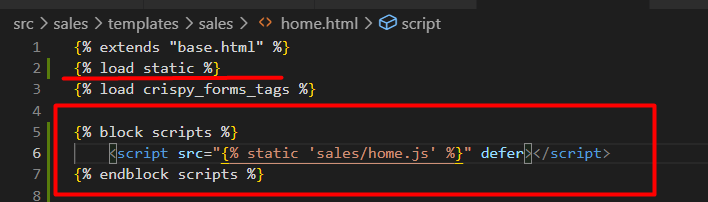


# Hello world from the console

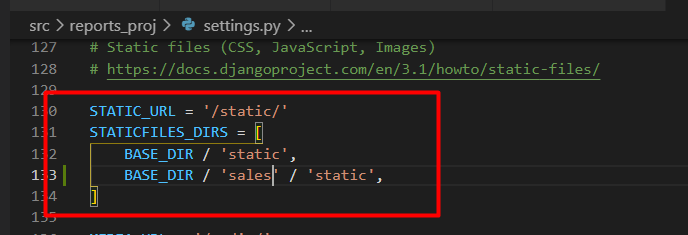
1. Add home.js file



1. Load home.js in view

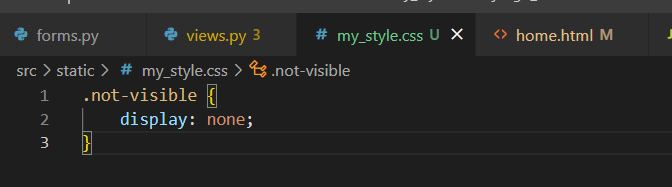


1. Add base dir static

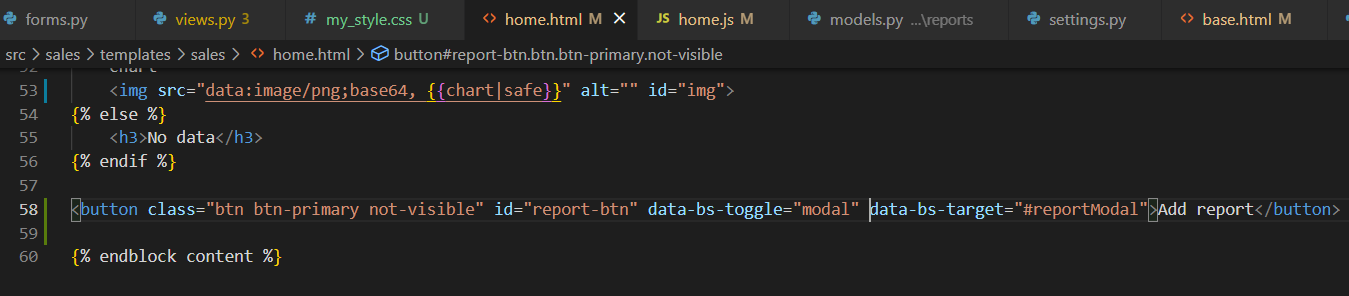


# Add the modal

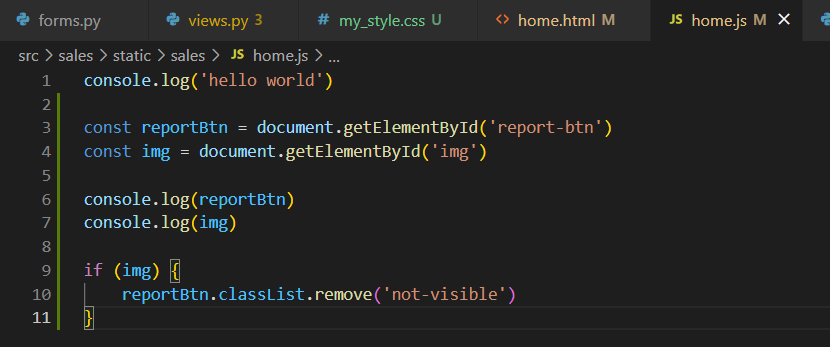
1. Css class



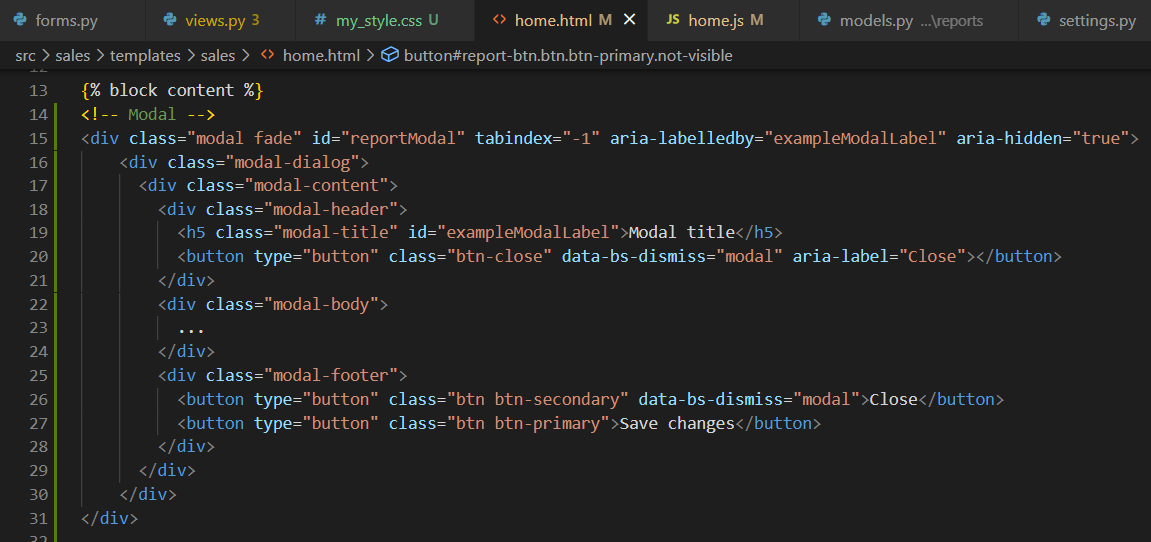
1. Add button “add report”



1. Write js file to only show add report button when chart is displayed

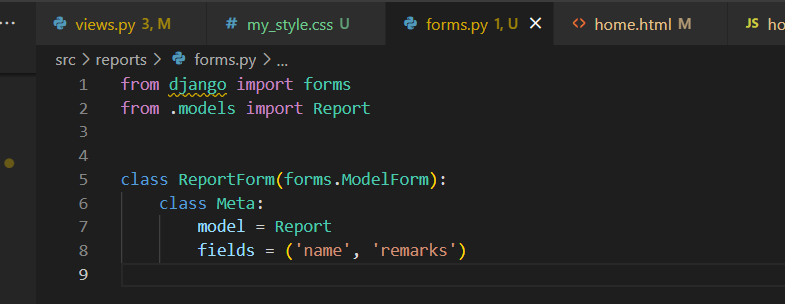


1. Add modal

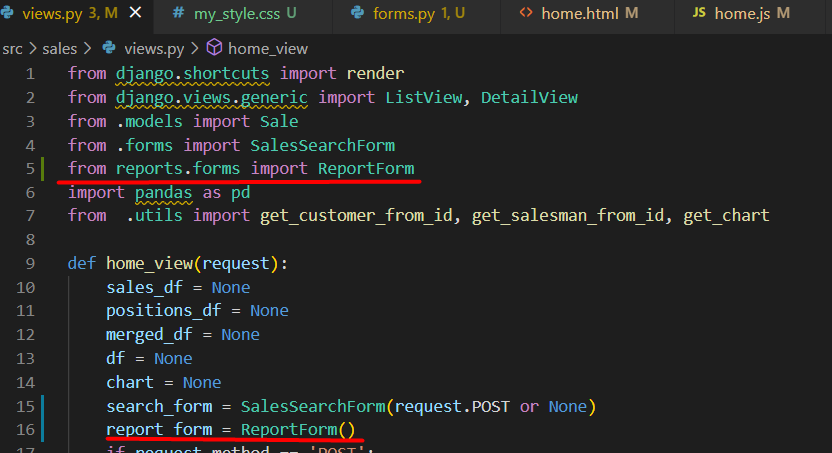


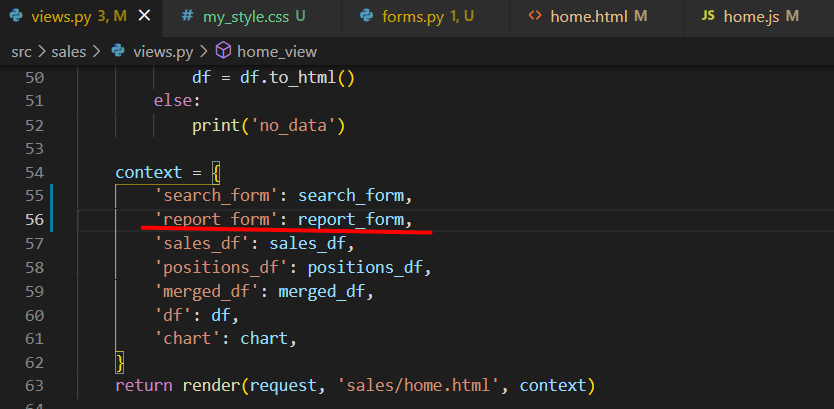
# Add report form to modal

1. Add ReportForm

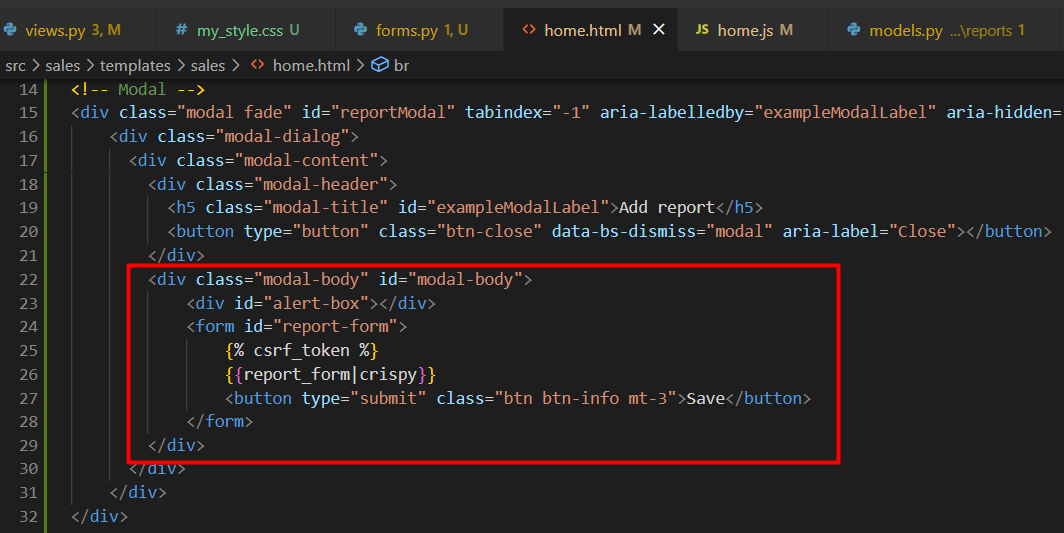


1. Import reportForm to views



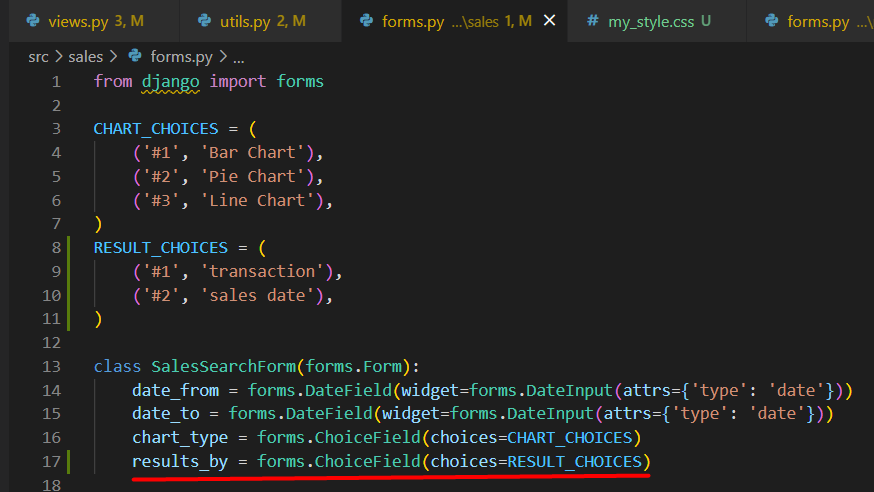


1. Display ReportForm to modal view

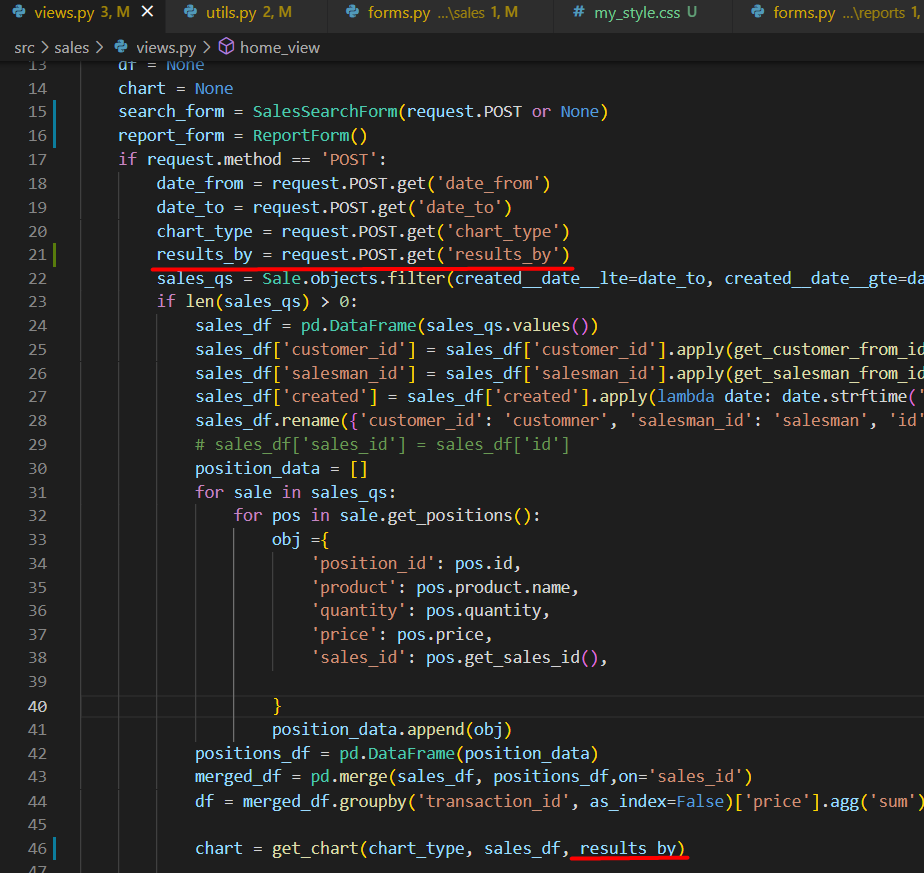


# Add result by fields

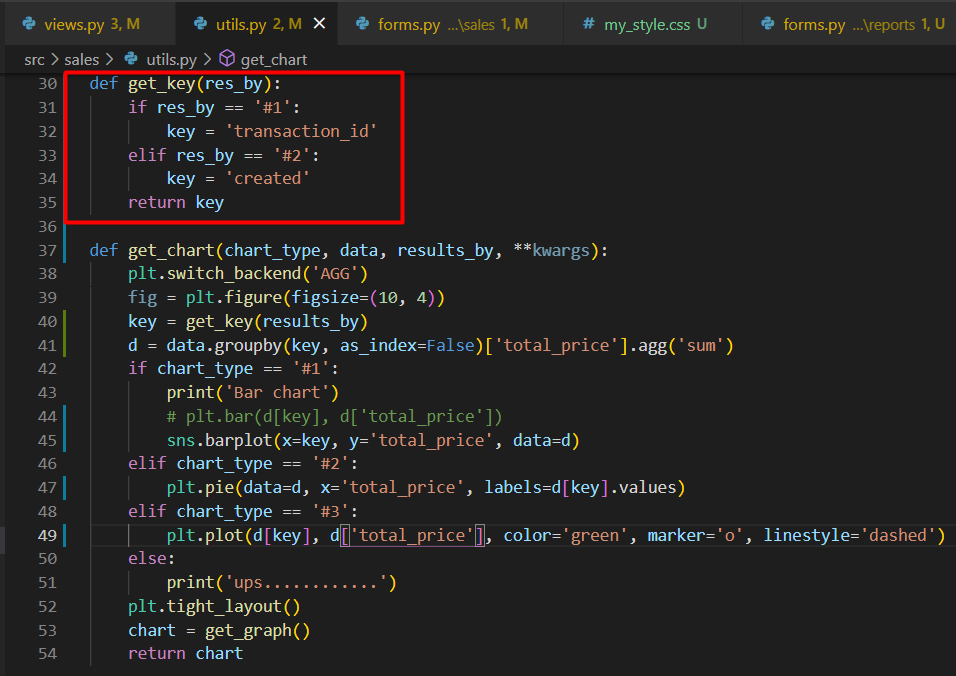
1. Add choice field to SaleSearchForm



1. Get field result by from form and send to get\_chart func

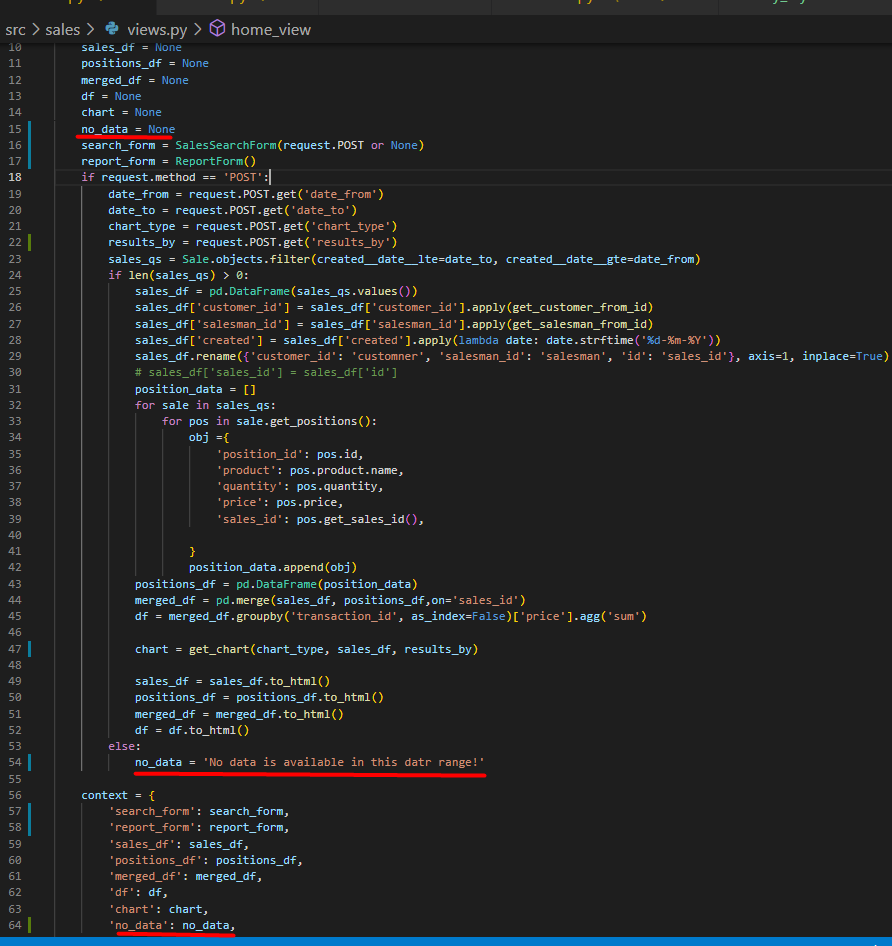


1. Change get\_chart func

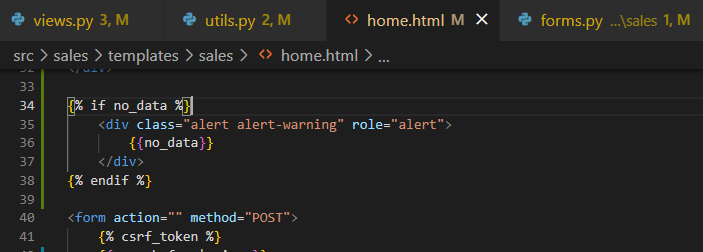


# No data available alert

1. Add no\_data to func view

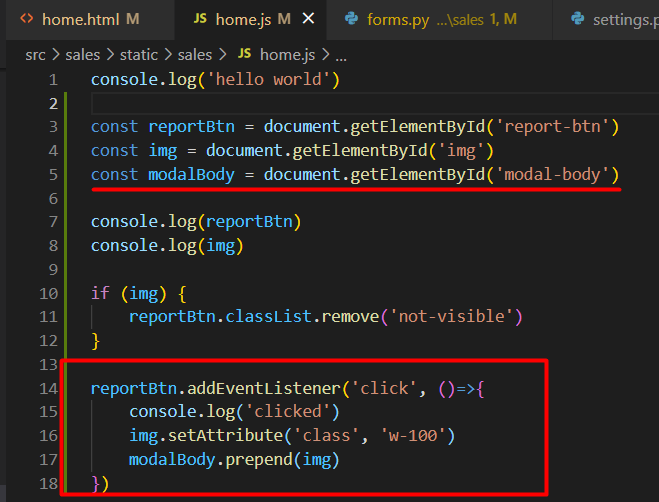


1. Add alert to view



# Add the chart to the modal

1. Add img report to the modal by js

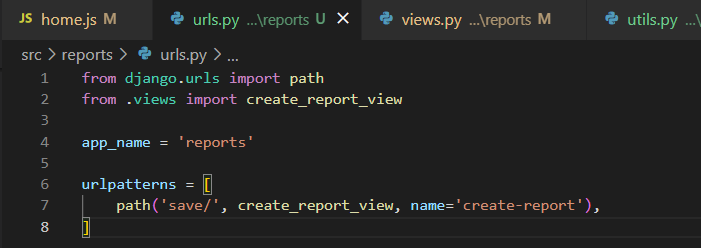


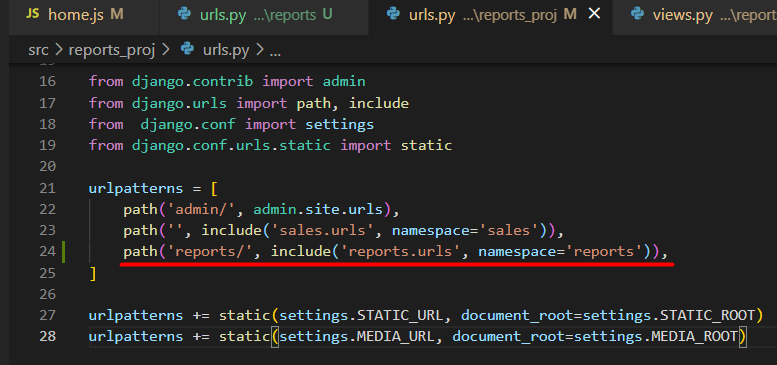
# Create report object

1. Create js func to send data to backend by ajax

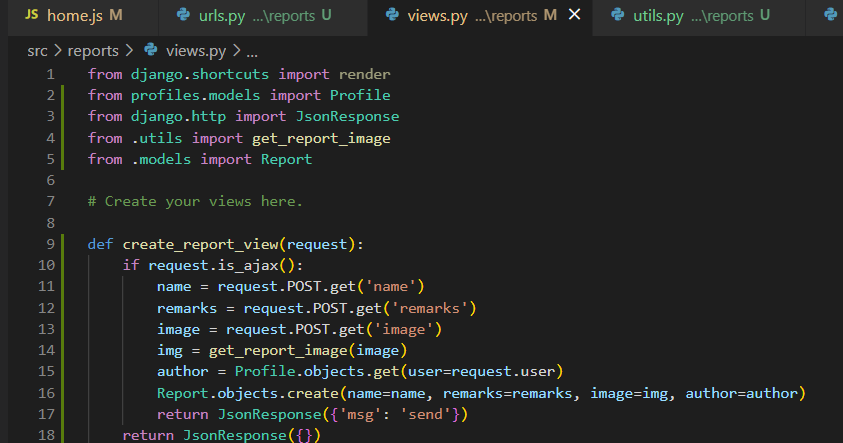


1. Create urls.py in reports app, and include it on urls.py of project

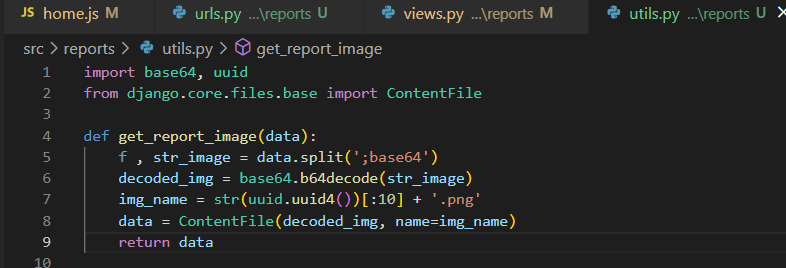




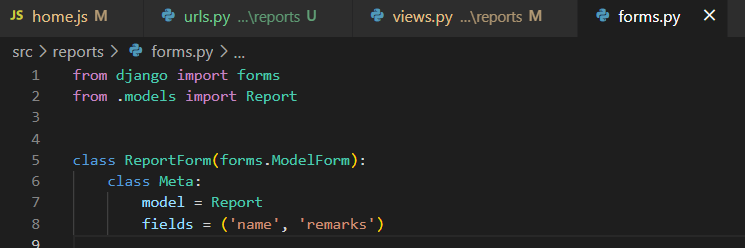
1. Create views func in views.py file

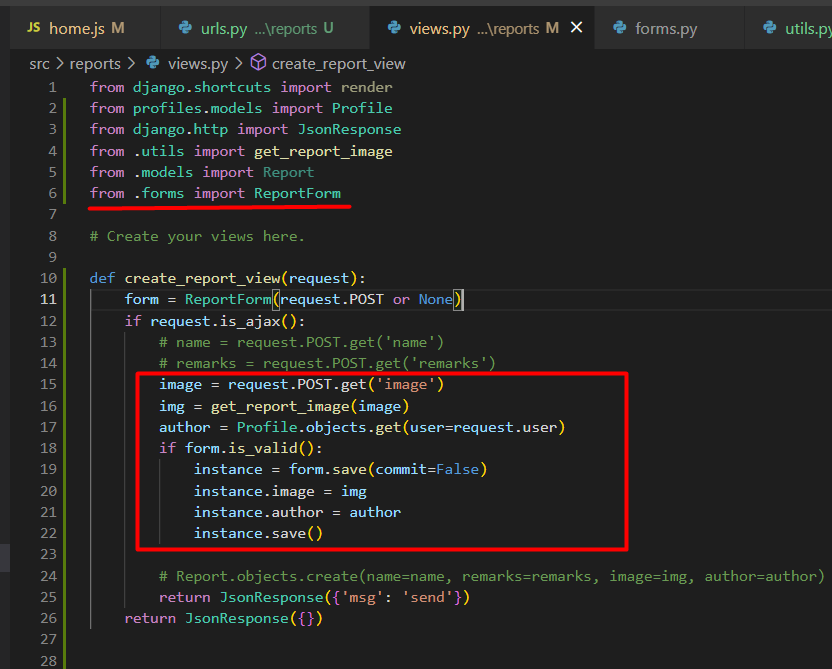


1. Create utils.py file to convert img data from ajax



1. Other way to create report by Form





# Adding alert to modal

1. Get alertBox from modal in js file and create handleAlert func

