

Mini Stock Market Django Tutorial

Yadira Taylor
yadira@lunacron.com

July 13, 2021

Setting up stockmarket project

Setup

- use requirements.txt file in source code
- In virtual env, `python -m django --version`
- **`django-admin startproject stockmarket`**
- `cd stockmarket`
- `python manage.py startapp quotes`

- views.py
- models.py
- admin.py
- tests.py
- create urls.py
- templates/quotes directory
- static/img directory
- copy pyladies3.jpg image to static/img directory

Project layout

- `stockmarket/settings.py`
- `stockmarket/urls.py`

- create about.html to quotes/templates/quotes
- edit views.py, and urls.py
- Inheritance of templates..
- Invoke: `python manage.py runserver`
- runs on port: 8000 (default)
- To test, browser: `http://localhost:8000`

Model: definition from djangoproject.com

A model is the single, definitive source of information about your data. It contains the essential fields and behaviors of the data you're storing. Generally, each model maps to a single database table.

- Create 2 models, Stock and HistoricalQuote for quotes app
- Note relationship **ForeignKey** in HistoricalQuote class
- `python manage.py makemigrations`
- `python manage.py migrate`
- Create superuser:
- `python manage.py createsuperuser`

- update admin.py in quotes to have both Historical quotes and stocks
- Verify that you see both objects: Historical quotes and Stocks
- Add these stocks:
- VZ Verizon, Telecommunication Services
- AAPL, Apple Inc., Information Technology
- INTC, Intel Co., Information Technology
- WMT, Walmart, Consumer Discretionary

Listing Stocks

- create templates/quotes/list_stocks.html
- edit urls.py
- edit views.py
- in base.html, add `{% url 'list_stocks' %}`

- create forms.py in quotes app
- Use forms.ModelForm when you work with models
- Update views.py, add functional view add_stock
- update urls, add new entry for adding stock
- `path('add_stock/', views.add_stock, name='add_stock')`

Data

- 1 Import Stock Data
- 2 Graph Data

- Create a custom Admin class
- Add filter
- Add date_hierarchy

Where to go next

- Create more than one application
- User and Role Management
- Class Based Views
- Explore more relationships between models: ManyToMany, OneToOneField
- File and ImageManagement
- Use postgresql or other database instead of default database
- More admin customizations
- Use more built-in tags and filters
- testing, integrate with pytest
- and many many more..