

Design, Development of FastAPI Application

Procedure:

1. Install dependencies
 2. Imports Libraries
 3. Create App
 4. Configure CORS to access API from anywhere
 5. Write Index Route - API for Hello World
 6. Run App
 7. Add addl. Web API Methods - Predict House Price API using
 - a. GET Method
 - b. POST Method
 8. Run FastAPI App
 9. Test FastAPI using
 - a. FastAPI Swagger API - <http://127.0.0.1:8000/docs>
 - b. Web Browser - 127.0.0.1:8000
-

1. Install uvicorn and fastapi

```
# pip install fastapi uvicorn
```

2. Imports Libraries

```
import uvicorn
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
import pickle
```

3. Create App

```
app = FastAPI()
```

4. Configure CORS to access API from anywhere

```
app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
```

5. Index Route – API for Hello World, opens automatically on <http://127.0.0.1:8000>

```
@app.get('/')
def index():
    return {'message': 'Hello World'}
```

```

# 6. Run App
if __name__ == '__main__':
    uvicorn.run(app, port=8080, host='0.0.0.0')
# 7a. Using GET Method
@app.get("/predictPrice")
def getPredictPrice(Area: int, BedRooms: int, BathRooms: int):
    rgModel = pickle.load(open("reg.pkl", "rb"))

    prediction = rgModel.predict([[Area,BedRooms,BathRooms]])

    return {
        'Price': prediction[0]
    }

# 7b. Using POST Method
from pydantic import BaseModel

class house(BaseModel):
    Area: int
    BedRooms: int
    BathRooms: int

@app.post("/predict")
def predictHousePrice(data: house):
    rgModel = pickle.load(open("reg.pkl", "rb"))

    data = data.dict()
    prediction = rgModel.predict([[data["Area"],data["BedRooms"],data["BathRooms"]]])

    return {
        'Price': prediction[0]
    }

# 8. Run the API with uvicorn with Reload Option - Auto Run after
edit source code
# uvicorn app:app --reload

# 9. Test API from Web Browser
#
http://127.0.0.1:8000/predictPrice?Area=1400&BedRooms=3&BathRooms=3

```