Flask is most used **framework** for making **API** for `machine learning` and `deep learning` applicaions.

To create API we need two parts

1. **Wrap the model**: code that *deal* with *model* and *return response*.
2. **Building the app**: This is where we `communicate with the client` and build an actual API with `Flask`.

To deploy api to be accessed using end users we use Heroku that need :-

- **Procfile** :- configuration file.
- **main**:- code of api, define routes and its functions.
- **requirements.txt**:- contains list of packages needed in code.
- **model.pkl** :- serialized pre-trained model.

Flask code :-

```
from flask import Flask , request , jsonify
from sklearn.externals import joblib
import numpy as np


app = Flask(__name__)

# main path (root )
@app.route("/")
def hello():
    return "Hello Every One To Qurany App"


@app.route('/uploadfile',methods=['GET','POST'])
def uploadfile():
    // do logic here, load model, predict values.
    return "Result"


if __name__ == '__main__':
    app.run(debug = True , port= 5874)
```

it consists from

1. `import` statments.
2. creating `instance` from flask app.
3. define **end points** using `@app.route(path)`.

end point with more details

- First **define** end point (**route**) using `@app.route(path)`
- add to **route** definition ttype of requests such as `GET`, `POST`, ..etc.
- create function and return response.
- receive data from client:
    - Form-url encoded :-

```
request.form.get("file")
```

    - raw json:- `request.get_josn["file"]`
    - files :- `file = request.files['file']`
- return response as json using **jsonfy** by using

```
return jsonify( result = str(file.filename))
```