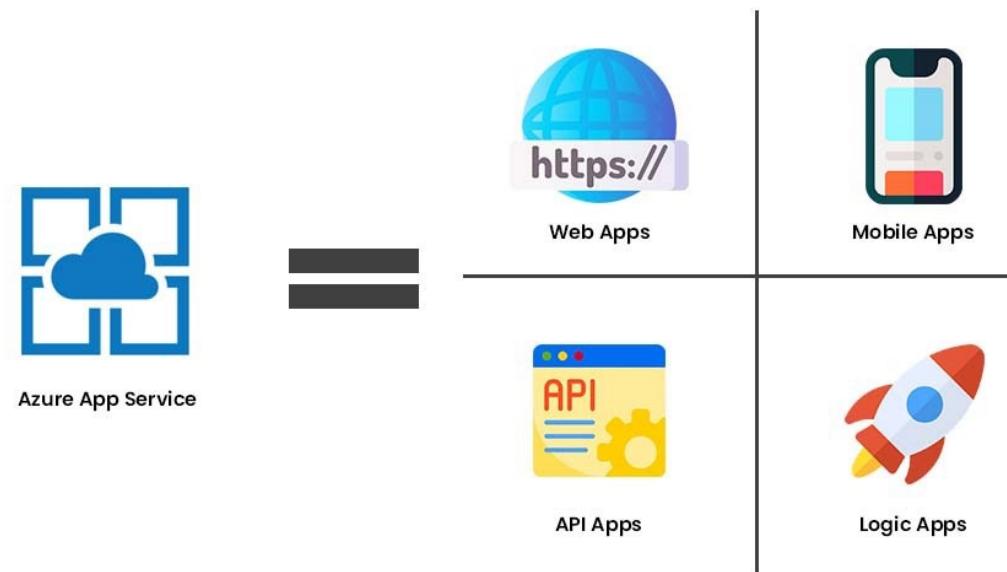
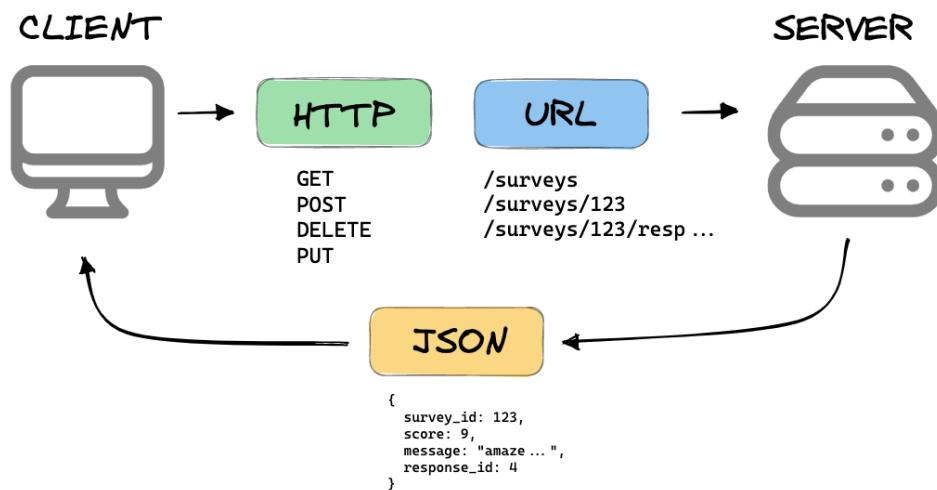
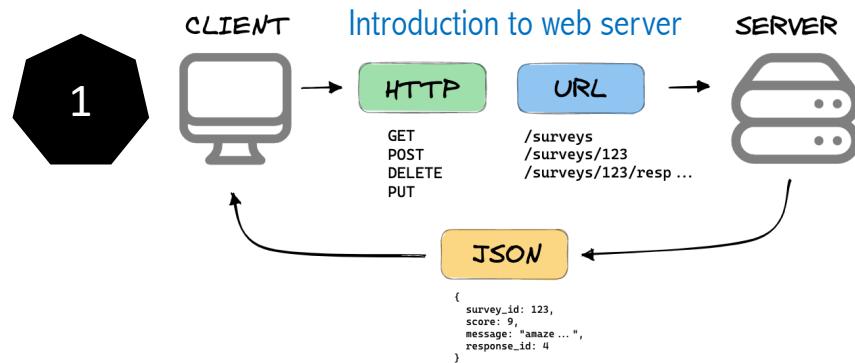


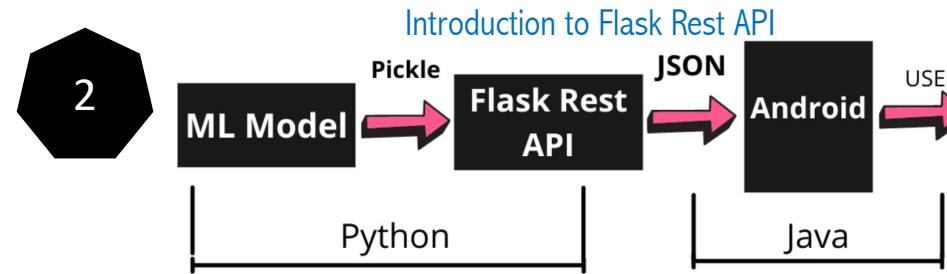
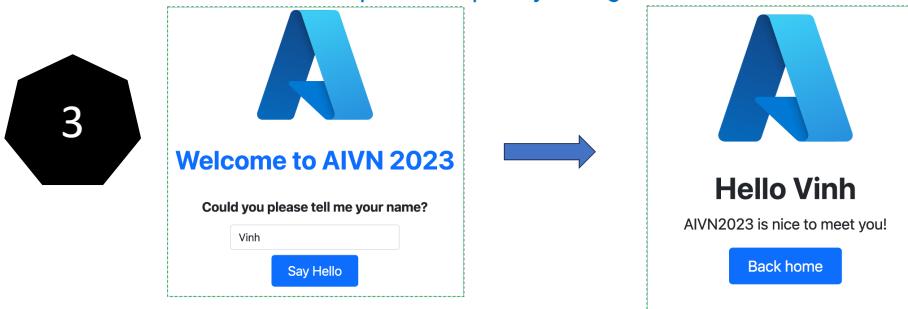
Deploy ML Models via a Web API On Local, Azure and PythonAnywhere Services By Examples



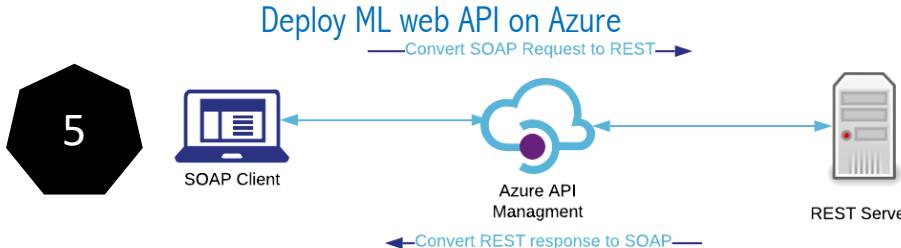
Objective



Simple example by using client and server



Deploy simple ML web API on local server



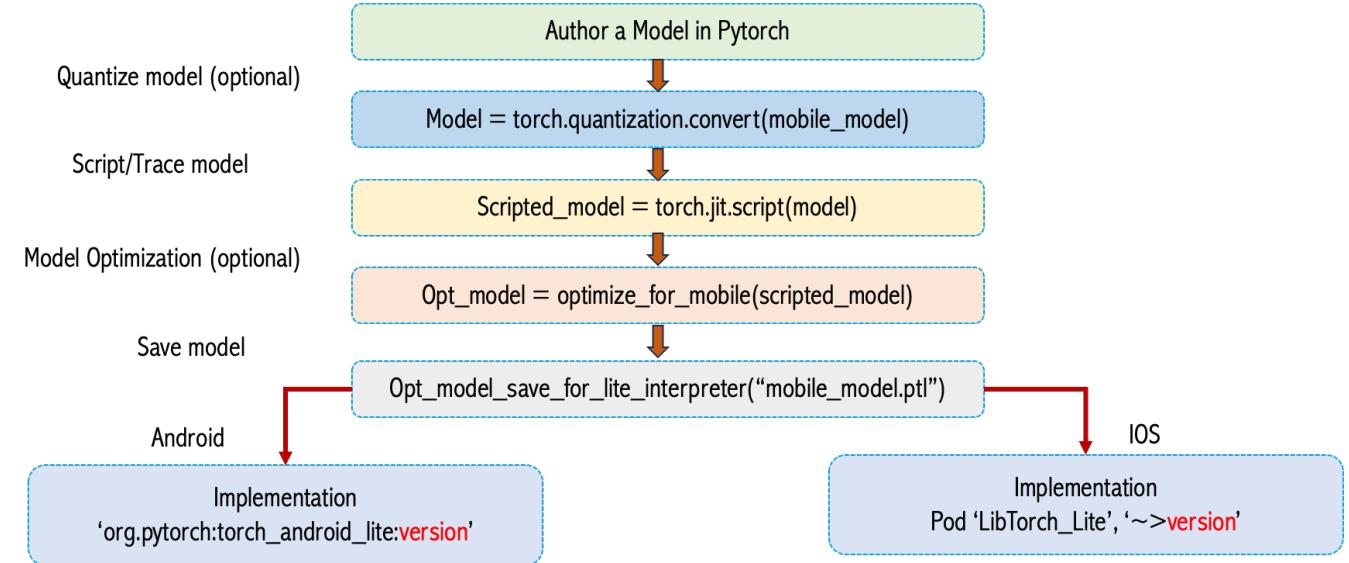
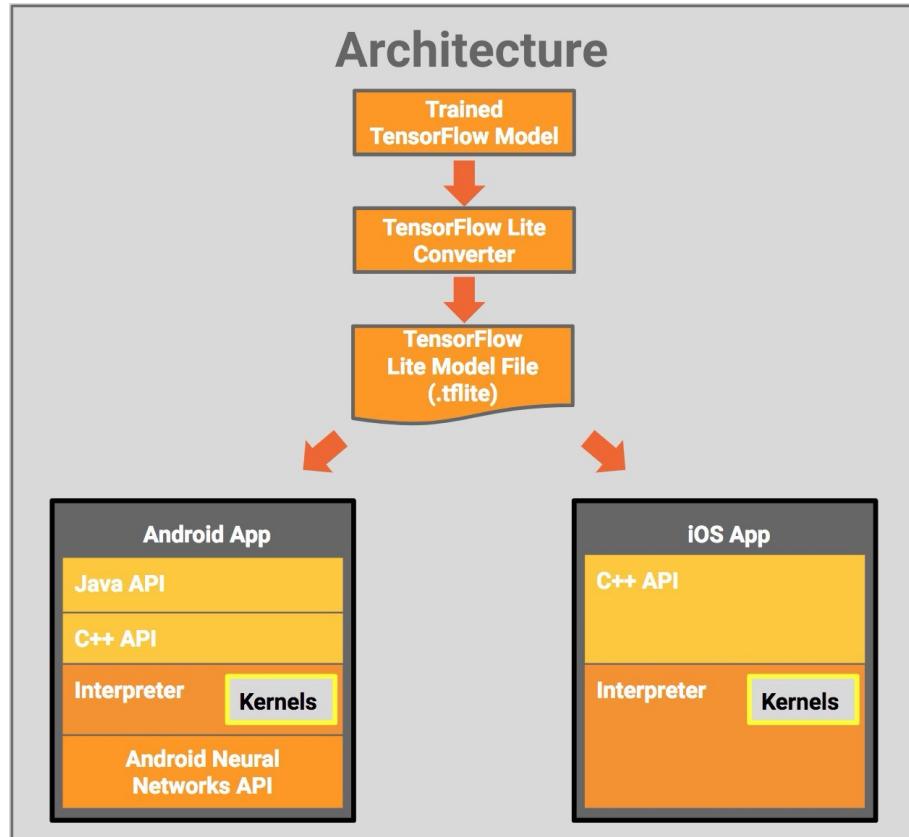
Outline

- Limitations of Traditional Mobile-based ML Apps
- What is a Rest API
- Deploy ML API on Flask Server
- Using ML API from Mobile Apps
- Deploy ML API on Azure Service
- Deploy ML API on PythonAnyWhere Service

Outline

- Limitations of Traditional Mobile-based ML Apps
- What is a Rest API
- Deploy ML API on Flask Server
- Using ML API from Mobile Apps
- Deploy ML API on Azure Service
- Deploy ML API on PythonAnyWhere Service

ML App on Mobile Device



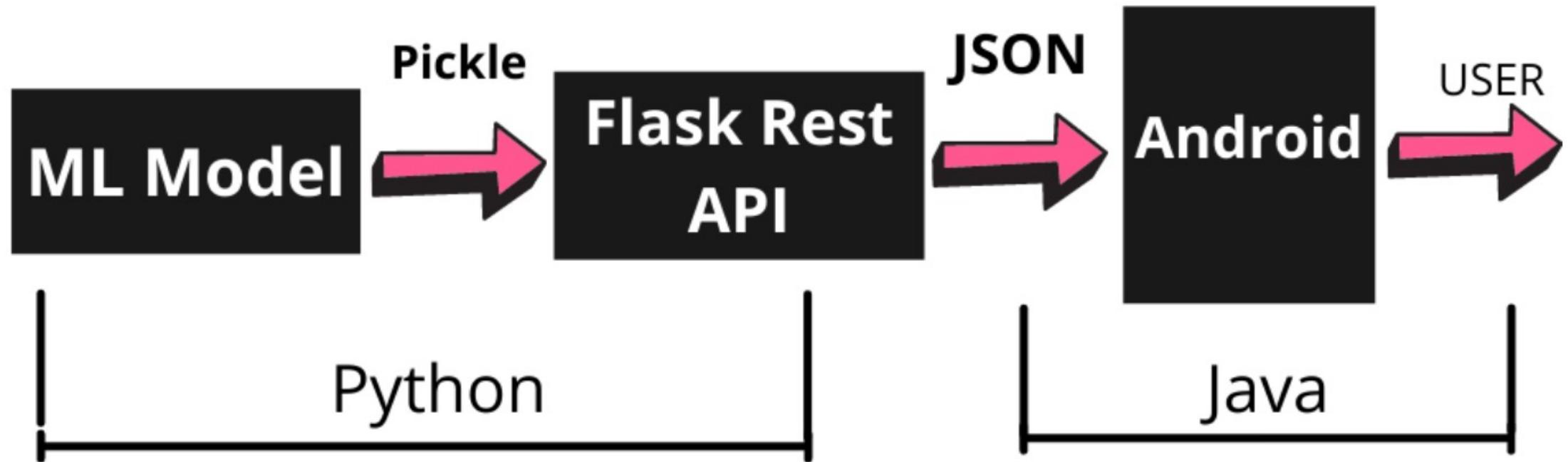
LIMITATION



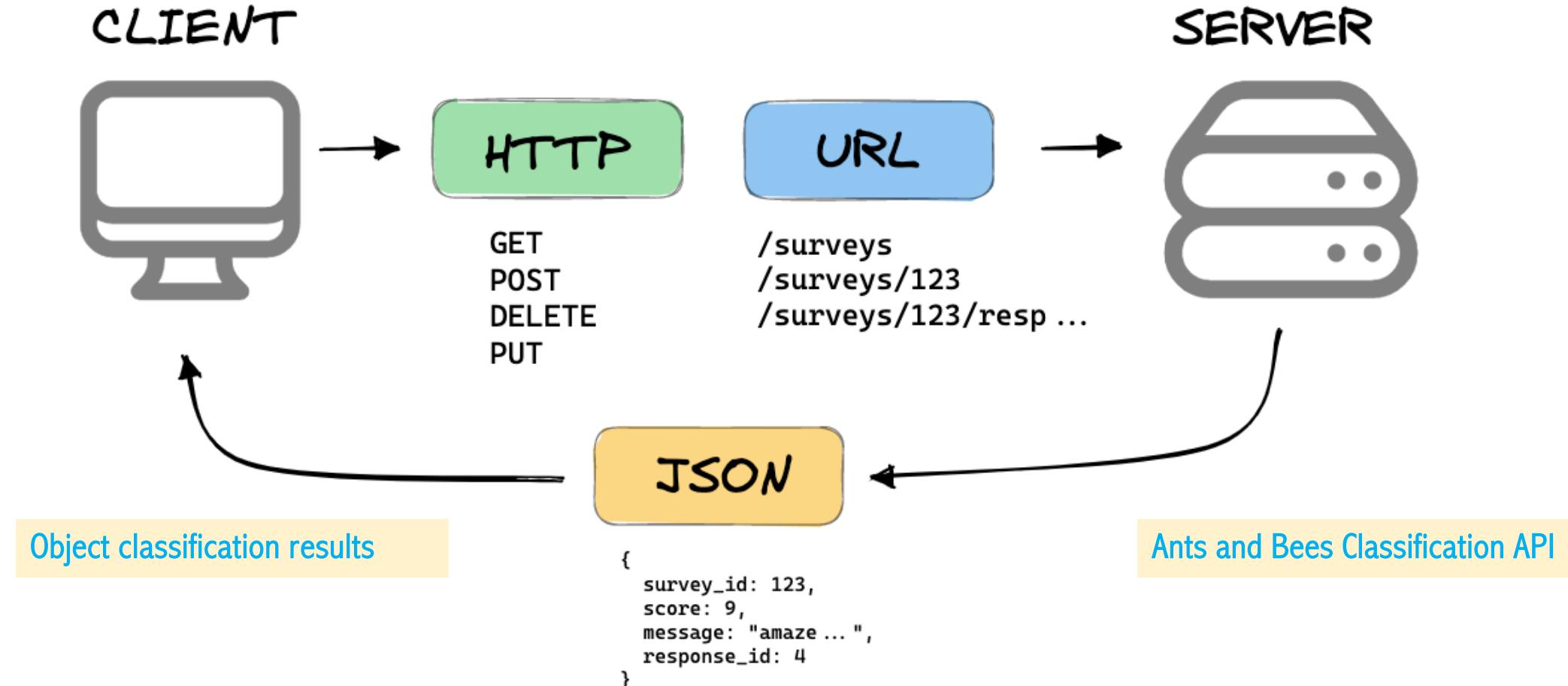
Outline

- Limitations of Traditional Mobile-based ML Apps
- What is a Rest API
- Deploy ML API on Flask Server
- Using ML API from Mobile Apps
- Deploy ML API on Azure Service
- Deploy ML API on PythonAnyWhere Service

What is a Rest API?



What is a Rest API?



What is a API?

URL endpoint: Is a URL link that represents the resources we want to access. Resources can be text, images, documents or any data entry.

An Application Programming Interface (API) is a way for two computers to talk to each other over the web.

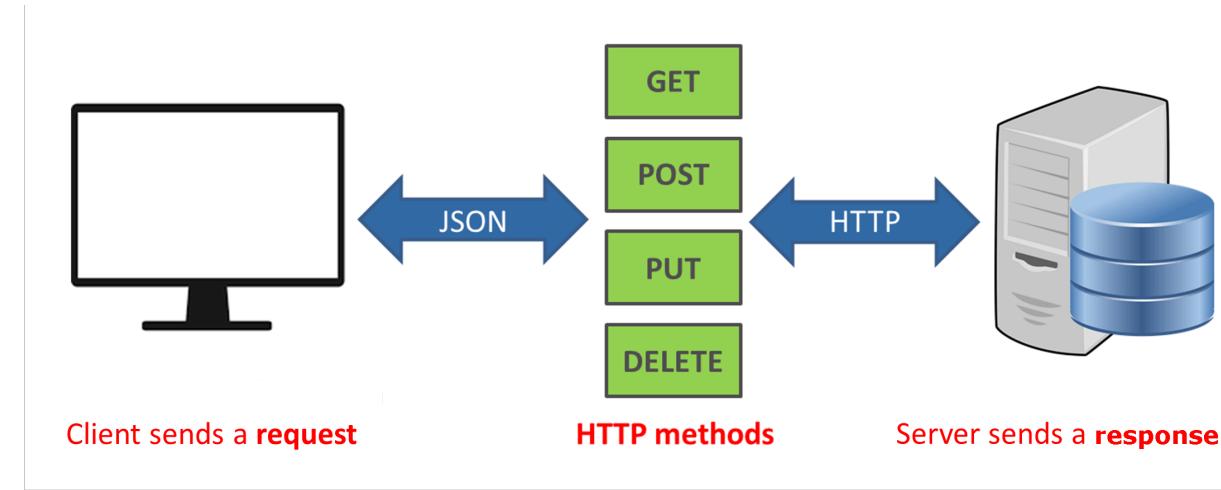
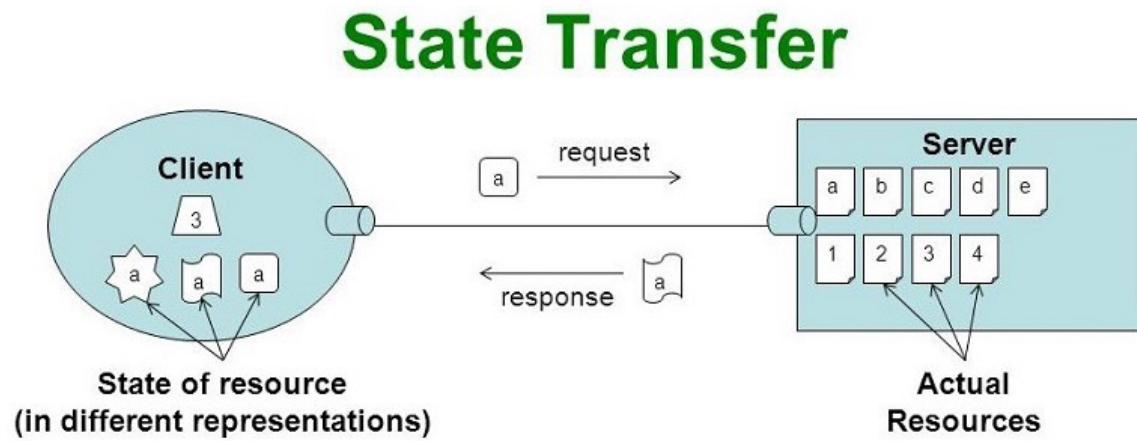
HTTP verb: Tells the server what we want to do with the URL endpoint resource. For example, a **POST** request means we want to create a new survey template and a **GET** request means we want to view an existing survey template.

Body message: Is an optional custom payload which contains a message with the properties and values we want to use to create or update a given resource.

Acronym	HTTP verb	URL endpoint resource	GET	POST	PUT	DELETE	Status code	Meaning
Create	POST	/surveys	Retrieve all surveys	Create a new survey	Bulk update surveys (not advised)	Remove all surveys (not advised)	200 OK	Request was successful.
Read	GET	/surveys/123	Retrieve the details for survey 123	Error	Update the details of survey 123 if it exists	Remove survey 123	301 Moved Permanently	For SEO purposes when a page has been moved and all link equity should be passed through.
Update	PUT & PATCH	/surveys/123/responses	Retrieve all responses for survey 123	Create a new response for survey 123	Bulk update responses for survey 123 (not advised)	Remove all responses for survey 123 (not advised)	401 Unauthorized	Server requires authentication.
Delete	DELETE	/responses/42	Retrieve the details for response 42	Error	Update the details of response 42 if it exists	Remove response 42	403 Forbidden	Client authenticated but does not have permissions to view resource.
							404 Not Found	Page not found because no search results or may be out of stock.
							500 Internal Server Error	Server side error. Usually due to bugs and exceptions thrown on the server side code.
							503 Server Unavailable	Server side error. Usually due to a platform hosting, overload and maintenance issue.

What is a Rest?

REST (REpresentational State Transfer)



Outline

- Limitations of Traditional Mobile-based ML Apps
- What is a Rest API
- Deploy ML API on Flask Server
- Using ML API from Mobile Apps
- Deploy ML API on Azure Service
- Deploy ML API on PythonAnyWhere Service

Define Object Classification Module

```
machine_learning_module.py ×

9  class CustomModel(torch.nn.Module):
10     def __init__(self, model_path=None):
11         super().__init__()
12
13         # Building the model
14         self.model = torchvision.models.resnet18(weights='IMAGENET1K_V1')
15         for param in self.model.parameters():
16             param.requires_grad = False
17
18         num_ftrs = self.model.fc.in_features
19         self.model.fc = nn.Linear(num_ftrs, 2)
20
21         weights = torch.load(model_path, map_location=torch.device('cpu'))
22         self.model.load_state_dict(weights)
```

Define Flask Server Module

```
transfer_learning_server.py x
8 app = Flask(__name__)
9
10
11 @app.route('/predict', methods=['POST'])
12 def predict():
13     if request.method == 'POST':
14         file = request.files['file']
15         img_bytes = file.read()
16         tensor = transforms.ToTensor()(Image.open(io.BytesIO(img_bytes)))
17         class_result = machine_learning_module.predict_by_image(tensor)
18         return jsonify({'class_name': class_result})
19
20
21 > if __name__ == '__main__':
22     app.run()
```

Flask Server Deployment

The screenshot shows the PyCharm IDE interface. At the top, the 'Run' tab is active, displaying the output of a Flask application run:

```
Run: Flask (transfer_learning_server.py) ×
FLASK_APP = transfer_learning_server.py
FLASK_ENV = development
FLASK_DEBUG = 0
In folder /Users/nguyendinhvinh2004@gmail.com/PycharmProjects/transfer_learning_server
/Users/nguyendinhvinh2004@gmail.com/PycharmProjects/transfer_learning_server/venv/bin/py
* Serving Flask app 'transfer_learning_server.py'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

In the main editor area, the file 'client.py' is open, containing the following code:

```
1 import requests
2
3 resp = requests.post("http://127.0.0.1:5000/predict",
4                      files={"file": open('ants.jpg', 'rb')})
5
6 print(resp.json())
```

At the bottom, the 'Run' tab is active again, showing the command to run the 'client' script:

```
Run: client ×
/Users/nguyendinhvinh2004@gmail.com/PycharmProjects/transfer_learning_s
/Users/nguyendinhvinh2004@gmail.com/PycharmProjects/transfer_learning_s
    warnings.warn(
{'class_name': 'Ants'}
```

Sending and Receiving Request Using Postman

The screenshot shows the Postman application interface. The URL in the header is `http://127.0.0.1:5000/predict`. The method is set to `POST` and the target URL is `http://0.0.0.0:5000/predict`. The `Body` tab is selected, showing the `form-data` option is chosen. A table under the body section contains one row with a checked checkbox in the first column, labeled `file`, and the value `ants.jpg` in the second column. Below this table is another row with columns for `Key`, `Type`, and `Value`. The `Body` tab also displays the response headers and results. The response shows a status of `200 OK` with a response time of `190 ms` and a size of `186 B`. The response body is a JSON object with the key `class_name` and value `"Ants"`.

Deploy ML model as API

1	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
2	6	148	72	35	0	33.6	0.627	50	1
3	1	85	66	29	0	26.6	0.351	31	0
4	8	183	64	0	0	23.3	0.672	32	1
5	1	89	66	23	94	28.1	0.167	21	0
6	0	137	40	35	168	43.1	2.288	33	1
7	5	116	74	0	0	25.6	0.201	30	0
8	3	78	50	32	88	31	0.248	26	1
9	10	115	0	0	0	35.3	0.134	29	0
10	2	197	70	45					
11	8	125	96	0					
12	4	110	92	0					
13	10	168	74	0					
14	10	139	80	0					
15	1	189	60	23					
16	5	166	72	19					

ML model Development

```
X = diabetes_dataset.drop(columns = 'Outcome', axis=1)
Y = diabetes_dataset['Outcome']

X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2,
classifier = svm.SVC(kernel='linear'))

#training the support vector Machine Classifier
classifier.fit(X_train, Y_train)

filename = 'diabetes_model.aio2023'
pickle.dump(classifier, open(filename, 'wb'))
```

Deploy ML model as API

1	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
2	6	148	72	35	0	33.6	0.627	50	1
3	1	85	66	29	0	26.6	0.351	31	0
4	8	183	64	0	0	23.3	0.672	32	1
5	1	89	66	23	94	28.1	0.167	21	0
6	0	137	40	35	168	40.2	0.128	37	0
7	5	116	74	0	0	24.3	0.446	31	1
8	3	78	50	32	88	28.7	0.248	31	0
9	10	115	0	0	0	30.3	0.300	33	0
10	2	197	70	45	543	31.4	0.096	33	1
11	8	125	96	0	0	32.0	0.437	34	0
12	4	110	92	0	0	33.6	0.370	33	0
13	10	168	74	0	0	34.3	0.337	33	1
14	10	139	80	0	0	35.1	0.209	33	0
15	1	189	60	23	846	35.7	0.090	33	1
16	5	166	72	19	175	36.0	0.176	32	0

API Development

```
@app.post('/diabetes_prediction')
def diabetes_predd(input_parameters: model_input):
    input_data = input_parameters.json()
    input_dictionary = json.loads(input_data)

    preg = input_dictionary['pregnancies']
    glu = input_dictionary['Glucose']
    bp = input_dictionary['BloodPressure']
    skin = input_dictionary['SkinThickness']
    insulin = input_dictionary['Insulin']
    bmi = input_dictionary['BMI']
    dpf = input_dictionary['DiabetesPedigreeFunction']
    age = input_dictionary['Age']
```

Deploy ML model as API

1	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
2	6	148	72	35	0	33.6	0.627	50	1
3	1	85	66	29	0	26.6	0.351	31	0
4	8	183	64	0	0	23.3	0.672	32	1
5	1	89	66	23	94	28.1	0.446	31	0
6	0	137	40	35	168	43.1	0.248	39	0
7	5	116	74	0	0	25.6	0.419	34	0
8	3	78	50	32	88	31	0.351	29	0
9	10	115	0	0	0	35.3	0.377	33	0
10	2	197	70	45	543	30.5	0.351	34	0
11	8	125	96	0	0	0	0.388	33	0
12	4	110	92	0	0	37.6	0.337	33	0
13	10	168	74	0	0	38	0.446	34	0
14	10	139	80	0	0	27.1	0.248	33	1
15	1	189	60	23	846	30.1	0.351	34	0
16	5	166	72	19	175	25.8	0.248	32	0

API Testing

import requests

```
url = 'http://127.0.0.1:8000/diabetes\_prediction'
```

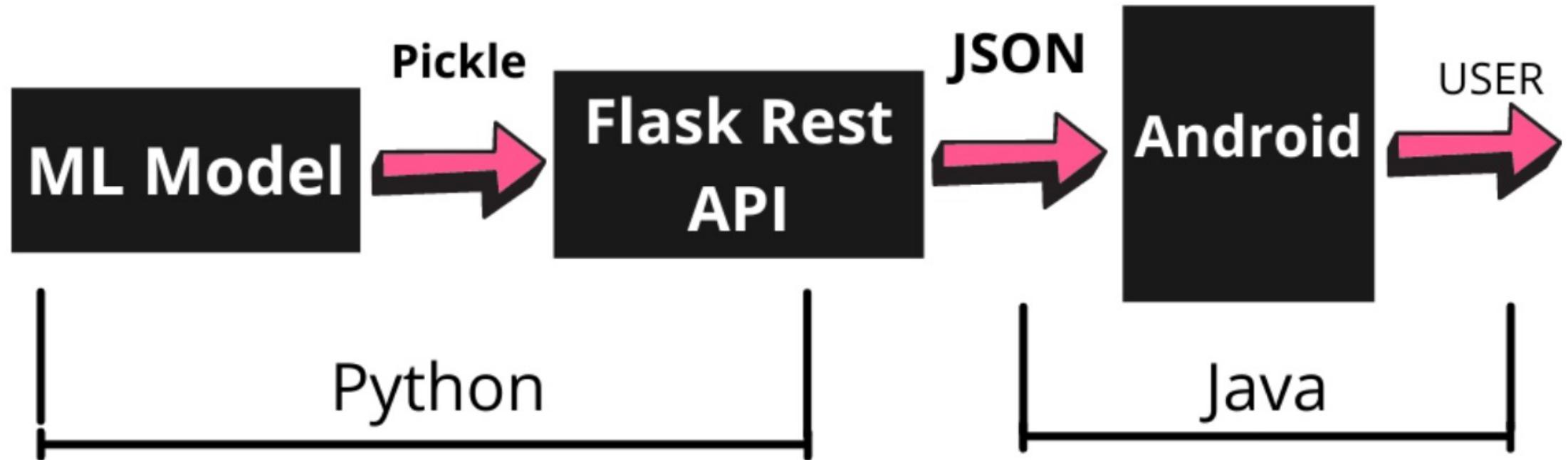
```
input_data_for_model = {
    'pregnancies': 5,
    'Glucose': 166,
    'BloodPressure': 72,
    'SkinThickness': 19,
    'Insulin': 175,
    'BMI': 25.8,
    'DiabetesPedigreeFunction': 0.587,
    'Age': 51}
```

}

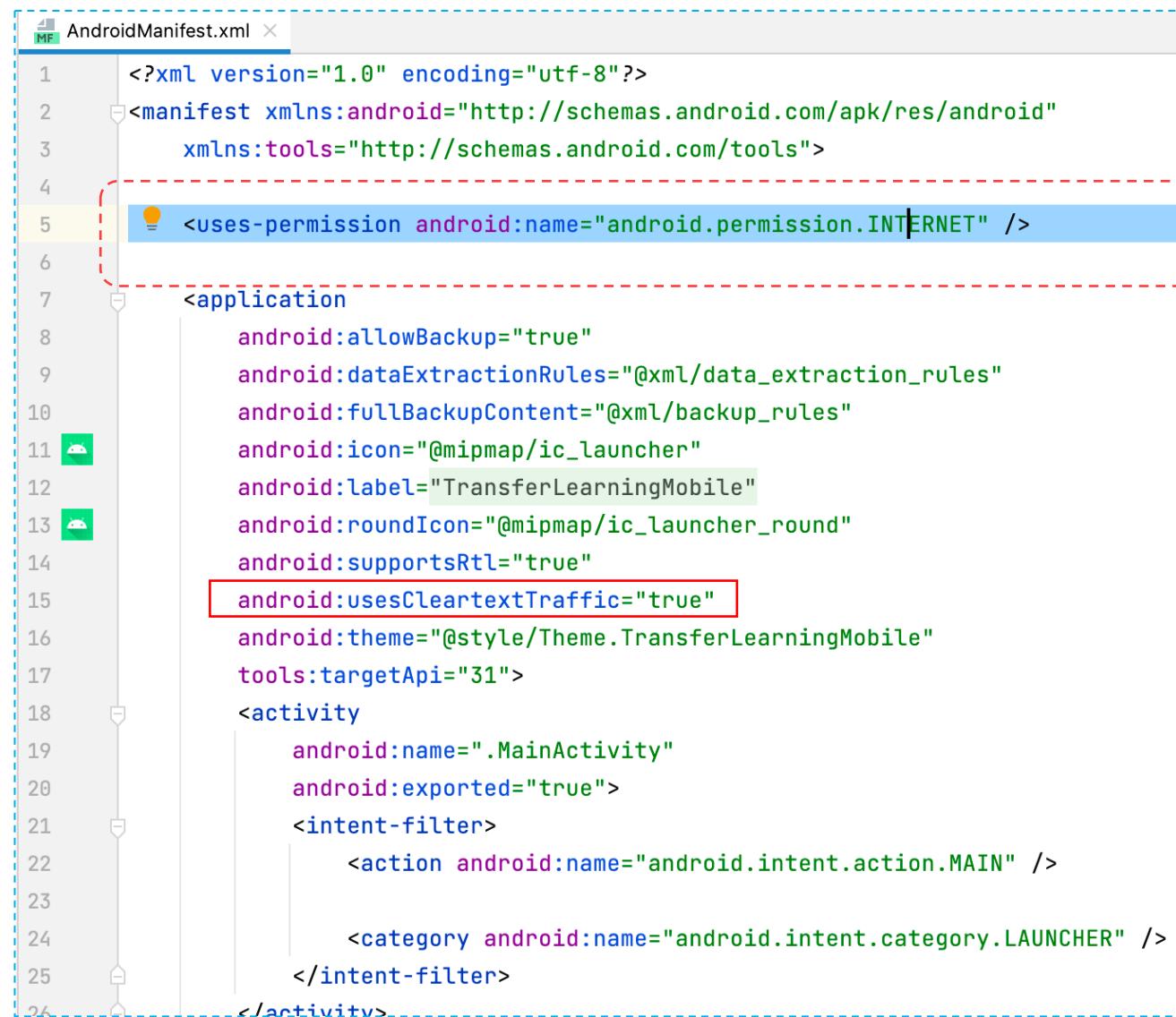
Outline

- Limitations of Traditional Mobile-based ML Apps
- What is a Rest API
- Deploy ML API on Flask Server
- Using ML API from Mobile Apps
- Deploy ML API on Azure Service
- Deploy ML API on PythonAnyWhere Service

Flask Rest API and Mobile



Internet Access Permission Configuration

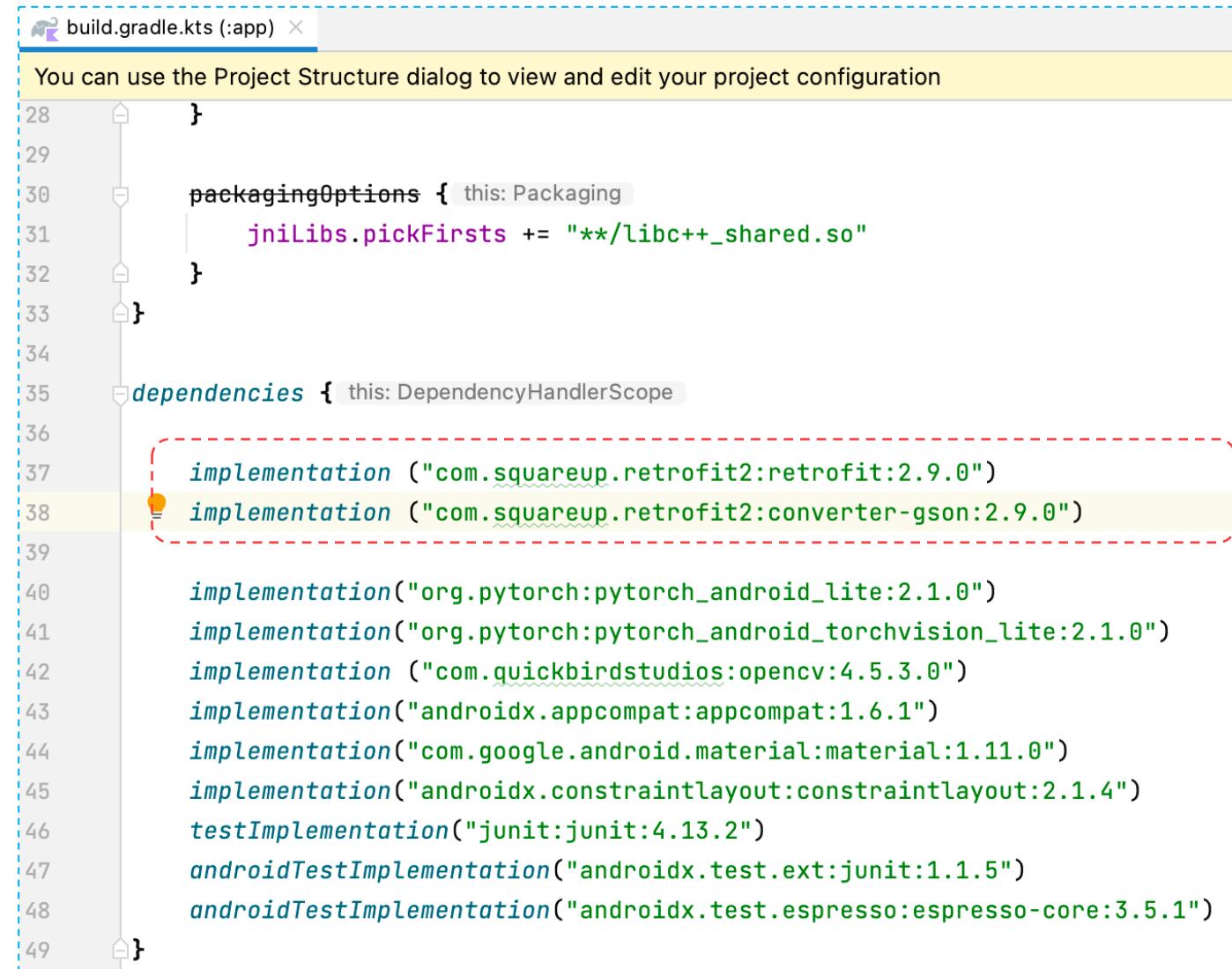


The screenshot shows the AndroidManifest.xml file in an IDE. A red dashed box highlights the permission declaration at line 5. A red rectangle highlights the `usesCleartextTraffic="true"` attribute at line 15. The XML code is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="TransferLearningMobile"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:usesCleartextTraffic="true"
        android:theme="@style/Theme.TransferLearningMobile"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Retrofit Configuration

Retrofit is an easy and fast library to retrieve and upload data via a REST-based web service



The screenshot shows the Android Studio code editor with the file `build.gradle.kts (:app)` open. The code is written in Kotlin. A red dashed rectangle highlights the `implementation` block for Retrofit dependencies at lines 37 and 38. The code is as follows:

```
28     }
29
30     packagingOptions {
31         this.jniLibs.pickFirsts += "**/libc++_shared.so"
32     }
33 }
34
35 dependencies {
36     implementation("com.squareup.retrofit2:retrofit:2.9.0")
37     implementation("com.squareup.retrofit2:converter-gson:2.9.0")
38
39     implementation("org.pytorch:pytorch_android_lite:2.1.0")
40     implementation("org.pytorch:pytorch_android_torchvision_lite:2.1.0")
41     implementation("com.quickbirdstudios:opencv:4.5.3.0")
42     implementation("androidx.appcompat:appcompat:1.6.1")
43     implementation("com.google.android.material:material:1.11.0")
44     implementation("androidx.constraintlayout:constraintlayout:2.1.4")
45     testImplementation("junit:junit:4.13.2")
46     androidTestImplementation("androidx.test.ext:junit:1.1.5")
47     androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
48
49 }
```

Retrofit Configuration

Result.java ×

```
1 package com.example.transferlearningmobile;
2
3 import ...
4
5 7 usages
6
7 public class Result {
8
9     2 usages
10    @SerializedName("class_name")
11    @Expose
12    private String class_name;
13
14    3 usages
15    public String getClass_name() {
16        return class_name;
17    }
18
19    no usages
20    public void setClass_name(String class_name) { this.class_name = class_name; }
```

ObjectDetectionService.java ×

```
1 package com.example.transferlearningmobile;
2
3
4 import ...
5
6 2 usages
7
8 public interface ObjectDetectionService {
9
10
11    1 usage
12    @FormUrlEncoded
13    @POST("/predict")
14    Call<Result> predict(@Field("file") String file);
15
16
17 }
```

Retrofit Configuration

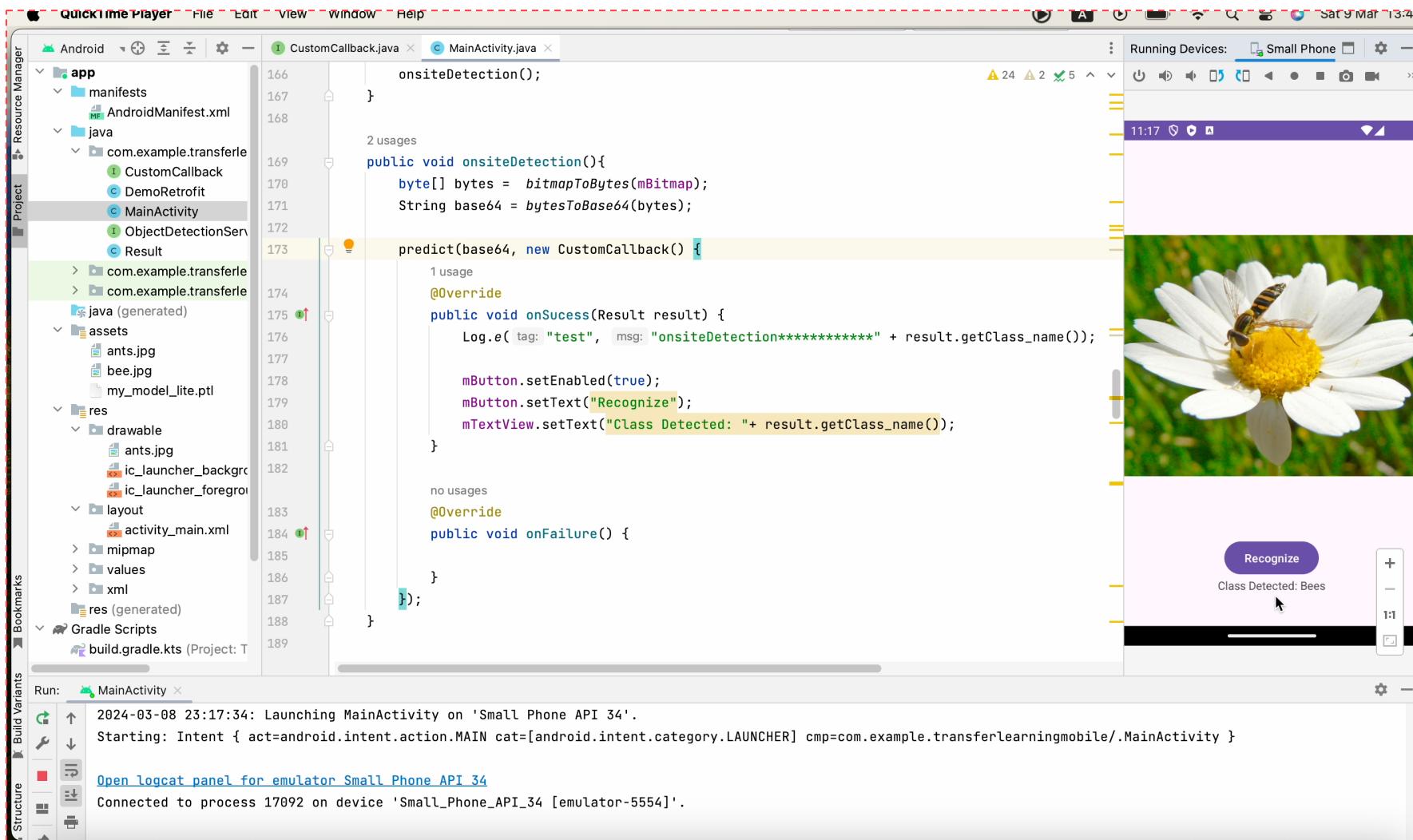
C DemoRetrofit.java ×

```
1 package com.example.transferlearningmobile;
2
3 import ...
4
5 1 usage
6 public class DemoRetrofit {
7
8     3 usages
9     public static Retrofit retrofit;
10
11    1 usage
12    public static Retrofit getInstance(){
13        if(retrofit == null){
14            retrofit = new Retrofit.Builder()
15                .baseUrl("http://10.0.2.2:5000/")
16                .addConverterFactory(GsonConverterFactory.create())
17                .build();
18        }
19        return retrofit;
20    }
21}
```

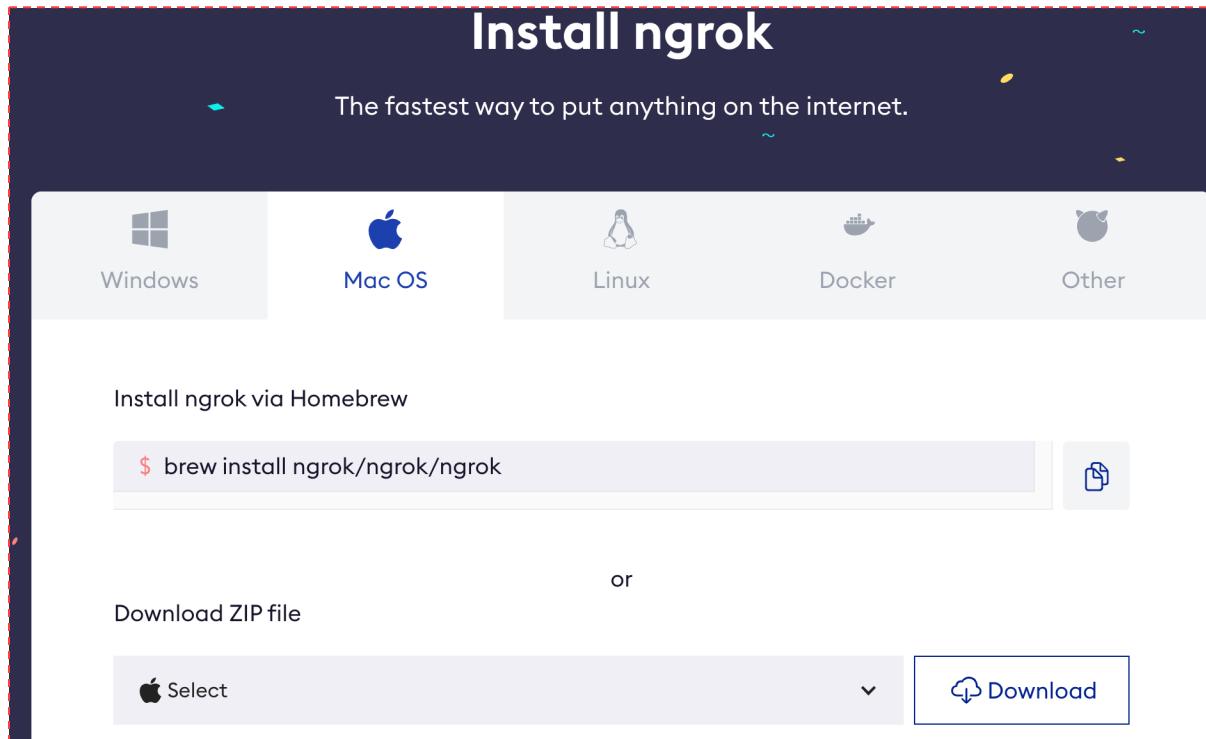
I CustomCallback.java ×

```
1 package com.example.transferlearningmobile;
2
3 import java.util.List;
4
5 2 usages 1 implementation
6
7 public interface CustomCallback {
8
9     1 usage 1 implementation
10    void onSucess(Result result);
11
12    no usages 1 implementation
13    void onFailure();
14
15}
```

Simulator Deployment: Flask Rest API and Mobile



Local Network Deployment: Flask Rest API and Mobile with ngrok



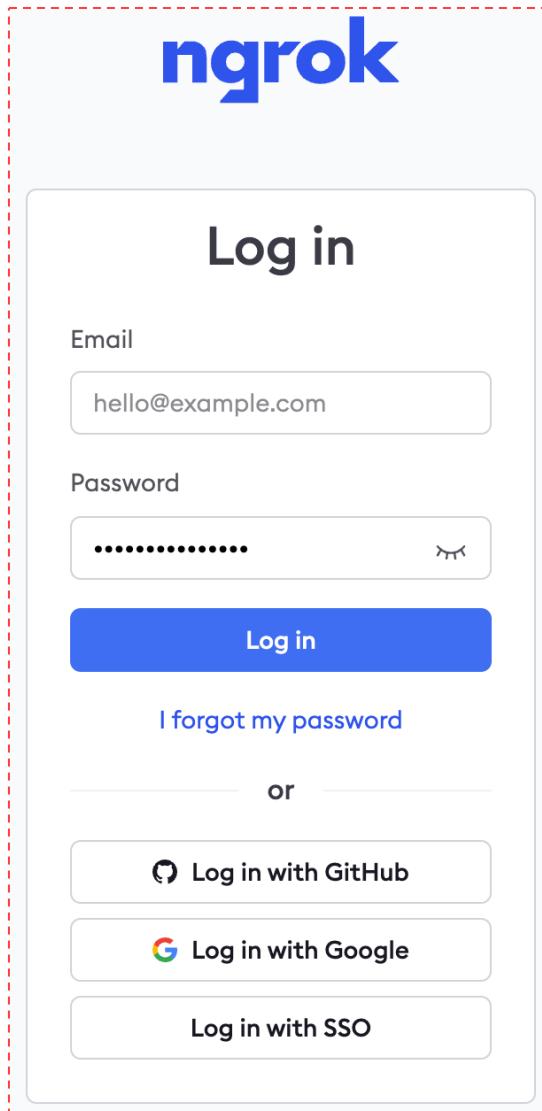
```
ngrok
Running `brew update --auto-update`...
Installing from the API is now the default behaviour!
You can save space and time by running:
  brew untap homebrew/core
==> Auto-updated Homebrew!
Updated 2 taps (dart-lang/dart and homebrew/core).
==> New Formulae
cmake-language-server

You have 7 outdated formulae installed.

==> Tapping ngrok/ngrok
Cloning into '/usr/local/Homebrew/Library/Taps/ngrok/homebrew-ngrok'...
remote: Enumerating objects: 142, done.
remote: Counting objects: 100% (82/82), done.
remote: Compressing objects: 100% (47/47), done.
remote: Total 142 (delta 27), reused 59 (delta 20), pack-reused 60
Receiving objects: 100% (142/142), 23.08 KiB | 2.88 MiB/s, done.
Resolving deltas: 100% (38/38), done.
Tapped 1 cask (13 files, 32.9KB).
==> Downloading https://bin.equinox.io/a/eZcX4as7yeX/ngrok-v3-3.6.0-darwin-amd64
#####
100.0%
```

<https://ngrok.com/download>

Create Account in Ngrok



Add auth token

```
$ ngrok config add-authtoken <token>
```



Don't have an authtoken? [Sign up](#)

ngrok config add-authtoken 69EL44epFLJ72q9H3esEP

```
nguyendinhvinh2004@gmail.com@NGUYENs-MacBook-Air-2 ~ % ngrok config add-authtoken  
[REDACTED] 69EL44epFLJ72q9H3esEP
```

```
Authtoken saved to configuration file: /Users/nguyendinhvinh2004@gmail.com/Library/Application Support/ngrok/ngrok.yml  
nguyendinhvinh2004@gmail.com@NGUYENs-MacBook-Air-2 ~ %
```

Setup Ngrok for Forwarding Request

```
ngrok http http://127.0.0.1:5000
```



The screenshot shows a terminal window with the following output:

```
ngrok http http://127.0.0.1:5000 - 125x24
ngrok
Try the new Traffic Inspector dev preview: https://ngrok.com/r/ti
Session Status          online
Account                  vinhnd (Plan: Free)
Version                 3.6.0
Region                  Asia Pacific (ap)
Latency                  -
Web Interface           http://127.0.0.1:4040
Forwarding              https://5362-2402-800-6347-13b5-b4ef-d105-e884-25df.ngrok-free.app -> http://localhost:5000
Connections             ttl     opn      rt1      rt5      p50      p90
                        0       0       0.00    0.00    0.00    0.00
```

The terminal window has a title bar "nguyendinhvinh2004@gmail.com — ngrok http http://localhost:5000 — 125x24". It also includes a status bar "(Ctrl+C to quit)".

The screenshot shows the Ngrok status page at `127.0.0.1:4040/status`. The top navigation bar includes links for The Old Reader, icc.skku.ac.kr/icc..., Reference cites, Others, English, Challenge Project, IR camera, Subpixel, Notice, Learning method, and Documentation. The main menu has tabs for ngrok (selected), online (highlighted in green), Inspect, and Status.

Configuration

Tunnels online - server 3.6.0

command_line	
URL	https://5362-2402-800-6347-13b5-b4ef-d105-e884-25df.ngrok-free.app
Addr	http://localhost:5000
Inspect	enabled
Proto	https

Global

Metrics

Connections

tunnel	total	open	/sec	1m	/sec	5m	/sec	15m
command_line	0	0	0.00	0.00	0.00	0.00	0.00	

Connection Durations (in seconds)

tunnel	50%	90%	95%	99%
command_line	0.00	0.00	0.00	0.00

Local Network Deployment: Flask Rest API and Mobile

The screenshot shows the Android Studio IDE interface. On the left, the code editor displays `DemoRetrofit.java` with the following content:

```
package com.example.transferlearningmobile;

import ...

1 usage

public class DemoRetrofit {

    3 usages
    8     public static Retrofit retrofit;

    1 usage
10    public static Retrofit getInstance(){
11        if(retrofit == null){
12            retrofit = new Retrofit.Builder()
13                .baseUrl("http://10.0.2.2:5000/")
14                .baseUrl("https://3be2-2402-800-6347-13b5-b4ef-d105-e884-25df.ngrok-free.app/")
15                .addConverterFactory(GsonConverterFactory.create())
16                .build();
17        }
18        return retrofit;
19    }
20}
21
22}
```

The code editor has tabs for `CustomCallback.java`, `DemoRetrofit.java` (which is selected), and `MainActivity.java`. Below the code editor, the status bar shows "MainActivity".

On the right, the "Running Devices" panel shows a Pixel 7 API 34 device. The app's user interface is displayed on the screen, featuring a large image of a flower with a bee in the center, and a purple "Recognize" button at the bottom right.

Local Network Deployment: Flask Rest API and Mobile

	One Discussion	Dashboard (7 requests)	25 Discussions
AsyncTask	941 ms	4,539 ms	13,957 ms
Volley	560 ms	2,202 ms	4,275 ms
Retrofit	312 ms	889 ms	1,059 ms

QUIZ TIME

Outline

- Limitations of Traditional Mobile-based ML Apps
- What is a Rest API
- Deploy ML API on Flask Server
- Using ML API from Mobile Apps
- Deploy ML API on Azure Service
- Deploy ML API on PythonAnyWhere Service

Web App in Flask

Local address: <http://127.0.0.1:5000/>



Welcome to AIVN 2023

Could you please tell me your name?

Say Hello



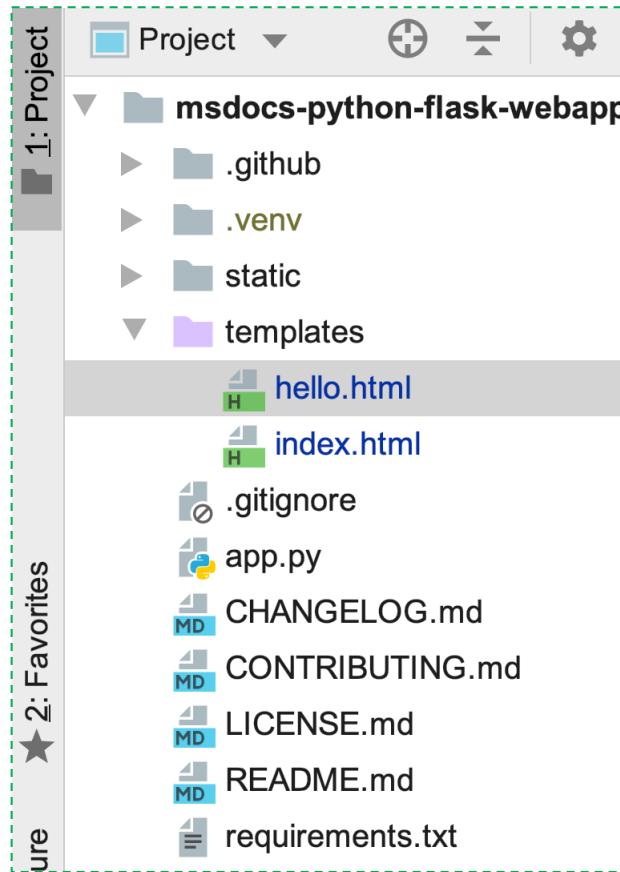
Hello Vinh

AIVN2023 is nice to meet you!

Back home

Web App in Flask

Step 1: Create simple Flask project



The screenshot shows a code editor with four tabs open, each with a red dashed border:

- app.py: Contains Python code for a Flask application. It defines a POST route for '/hello' that prints the name from the request form and returns a template named 'hello.html'. If no name is provided, it prints a message and redirects to the 'index' page.
- requirements.txt: A text file listing dependencies:
 - Flask==2.2.2
 - gunicorn
 - Werkzeug==2.2.2
- index.html: An HTML file with a title and a main section containing a centered image and text.
- hello.html: An HTML file with a title and a main section containing a centered image and text.

Web App in Flask

Step 2: Go to the application folder and Create a virtual environment for the app



Windows macOS/Linux

Cmd

```
py -m venv .venv  
.venv\scripts\activate
```

Windows macOS/Linux

Bash

```
python3 -m venv .venv  
source .venv/bin/activate
```

nguyendinhvinh2004@gmail.com@NGUYENs-MacBook-Air-2 simple_web_azure % python3 -m venv .venv
nguyendinhvinh2004@gmail.com@NGUYENs-MacBook-Air-2 simple_web_azure % source .venv/bin/activate

(.venv) nguyendinhvinh2004@gmail.com@NGUYENs-MacBook-Air-2 simple_web_azure %

Step 3: Install the dependencies

Console Copy

```
pip install -r requirements.txt
```

```
(.venv) nguyendinhvinh2004@gmail.com@NGUYENs-MacBook-Air-2 simple_web_aivn2023 % pip install -r requirements.txt

Requirement already satisfied: Flask==2.2.2 in ./venv/lib/python3.8/site-packages (from -r requirements.txt (line 1)) (2.2.2)
Requirement already satisfied: gunicorn in ./venv/lib/python3.8/site-packages (from -r requirements.txt (line 2)) (21.2.0)
Requirement already satisfied: Werkzeug==2.2.2 in ./venv/lib/python3.8/site-packages (from -r requirements.txt (line 3)) (2.2.2)
Requirement already satisfied: Jinja2>=3.0 in ./venv/lib/python3.8/site-packages (from Flask==2.2.2->-r requirements.txt (line 1)) (3.1.3)
Requirement already satisfied: itsdangerous>=2.0 in ./venv/lib/python3.8/site-packages (from Flask==2.2.2->-r r
```

Web App in Flask

Step 4: Run the app

Console

```
flask run
```

```
(.venv) nguyendinhvinh2004@gmail.com@NGUYENs-MacBook-Air-2 simple_web_aivn2023 % flask
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```



pattern korea The Old Reader icc.skku.ac.kr/icc_... Reference cites All Bookmarks



Welcome to AIVN 2023

Could you please tell me your name?

Say Hello

Web app On Azure

Step 1: to host your application in Azure, you need to create Azure App Service web app in Azure.



```
(.venv) nguyendinhvinh2004@gmail.com@NGUYENs-MacBook-Air-2 simple_web_aivn2023 % az login
A web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with `az login --use-device-code`.
```

You have logged into Microsoft Azure!

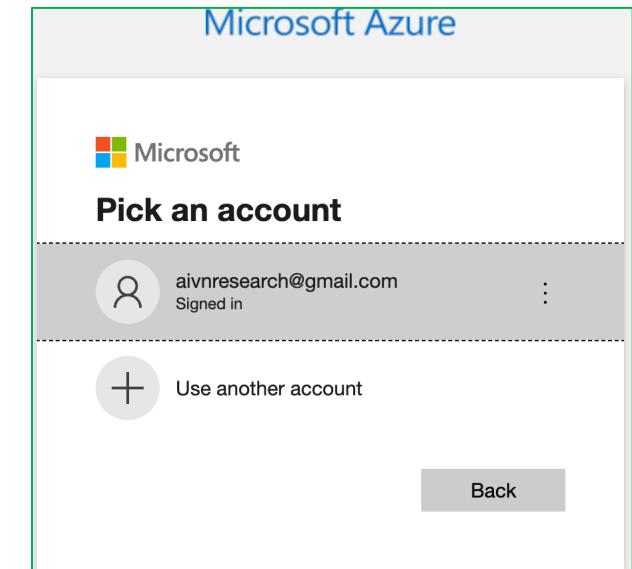
You can close this window, or we will redirect you to the [Azure CLI documentation](#) in 1 minute.

Announcements

[Windows only] Azure CLI is collecting feedback on using the [Web Account Manager](#) (WAM) broker for the login experience.

You may opt-in to use WAM by running the following commands:

```
az config set core.enable_broker_on_windows=true
az account clear
az login
```



How to install the Azure CLI

Install Azure CLI on macOS

Bash

```
brew update && brew install azure-cli
```

Copy

Install Azure CLI on Window

Microsoft Installer
(MSI)

Microsoft Installer (MSI) with
PowerShell

Windows Package
Manager

ZIP
Package

Latest version

Download and install the latest release of the Azure CLI. When the installer asks if it can make changes to your computer, select the "Yes" box.

[Latest MSI of the Azure CLI \(32-bit\)](#)

[Latest MSI of the Azure CLI \(64-bit\)](#)

If you have previously installed the Azure CLI, running either the 32-bit or 64-bit MSI will overwrite an existing installation.

Create a Web app in Azure

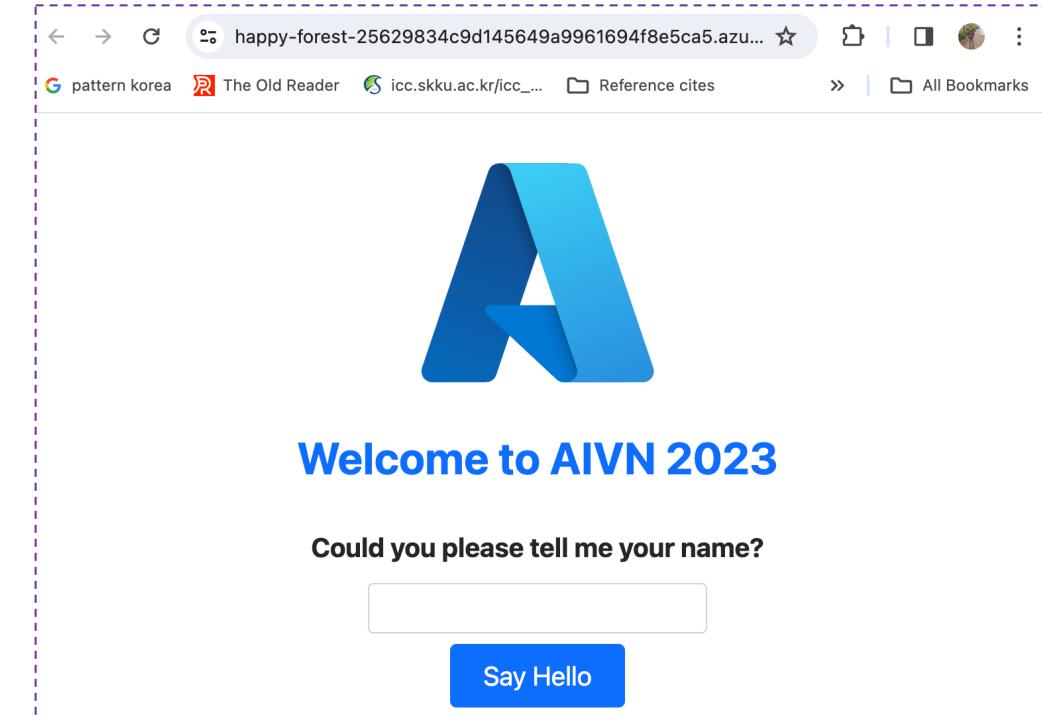
Step 2: Create the webapp and other resources, then deploy your code to Azure using [az webapp up](#).

Azure CLI

```
az webapp up --runtime PYTHON:3.9 --sku B1 --logs
```

Copy

```
Creating Resource group 'aivnresearch_rg_2200' ...
Resource group creation complete
Creating AppServicePlan 'aivnresearch_asp_1387' or Updating if already exists
 Readonly attribute name will be ignored in class <class 'azure.mgmt.web.v2023_01_01.models._models_py3.AppServicePlan'>
Creating webapp 'happy-forest-25629834c9d145649a9961694f8e5ca5' ...
Configuring default logging for the app, if not already enabled
Creating zip with contents of dir /Users/nguyendinhvinh2004@gmail.com/Desktop/New_Data/AI VN/AIO 2023/Web Module/simple_web_azure/simple_web_aivn2023 ...
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
Deployment endpoint responded with status code 202
```



The screenshot shows the Microsoft Azure portal interface for managing App Services. The top navigation bar includes the Microsoft Azure logo, a search bar, and various icons for account settings and notifications. The user is signed in as 'aivnresearch@gmail.com'.

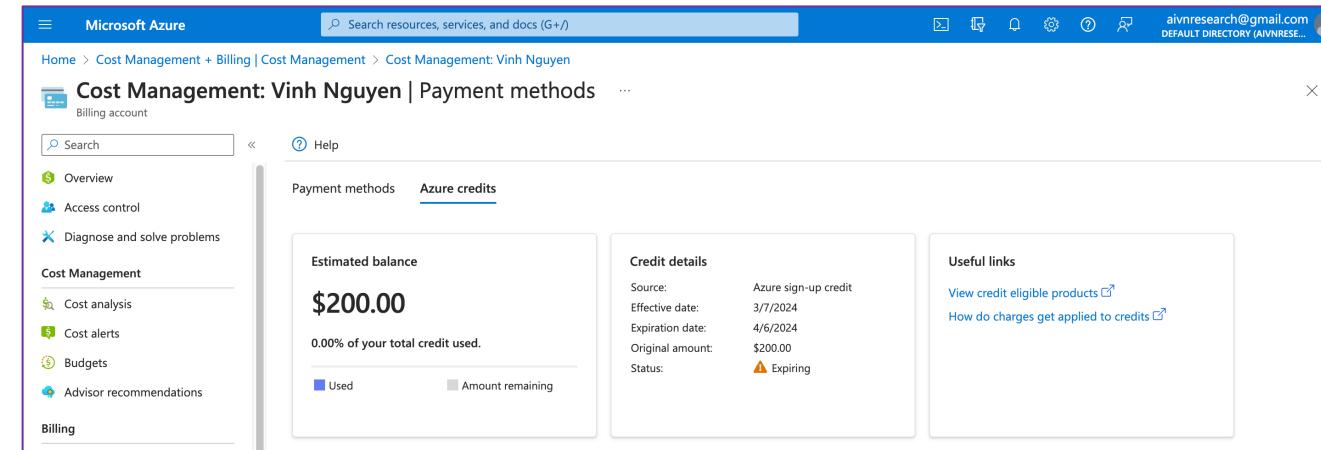
The main title is 'App Services' with a back arrow pointing to 'Home'. Below the title, it says 'Default Directory (aivnresearch@gmail.onmicrosoft.com)'. The toolbar below the title includes buttons for 'Create', 'Manage Deleted Apps', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Assign tags', 'Start', 'Restart', 'Stop', and 'Delete'.

Filtering options include 'Subscription equals all', 'Resource group equals all', 'Location equals all', and a button to 'Add filter'. There are also grouping and view selection buttons ('No grouping' and 'List view').

The table displays two records:

Name	Status	Location	Pricing Tier	App Service Plan	Subscription	App Type	More
black-plant-4c9bd9962e664726986146ce7ae479e4	Running	East US	Basic	aivnresearch_asp_3215	Azure subscription 1	Web App	...
happy-forest-25629834c9d145649a9961694f8e5c...	Running	East US	Basic	aivnresearch_asp_1387	Azure subscription 1	Web App	...

Web App Services Management on Azure



The screenshot shows the Microsoft Azure Cost Management portal. The main heading is "Cost Management: Vinh Nguyen | Payment methods". On the left, there's a sidebar with "Billing account" and "Cost Management" sections. The "Cost Management" section includes "Cost analysis", "Cost alerts", "Budgets", and "Advisor recommendations". The "Billing" section has a "Search" bar and a "Help" link. The "Payment methods" tab is selected, showing an "Estimated balance" of "\$200.00" and "0.00% of your total credit used". Below this, there are two bars: "Used" (blue) and "Amount remaining" (grey). To the right, the "Azure credits" tab is selected, displaying "Credit details" for an "Azure sign-up credit": Source: Azure sign-up credit, Effective date: 3/7/2024, Expiration date: 4/6/2024, Original amount: \$200.00, Status: Expiring. There are also "Useful links" for "View credit eligible products" and "How do charges get applied to credits".

Free Plan	Cores	RAM	Storage	Pay as you go
F1 Free	Shared (60 CPU minutes / day)	1 GB	1.00 GB	\$0

Basic Service Plan	Cores	RAM	Storage	Pay as you go
B1	1	1.75 GB	10 GB	\$0.018/hour
B2	2	3.50 GB	10 GB	\$0.035/hour
B3	4	7 GB	10 GB	\$0.07/hour

Outline

- Limitations of Traditional Mobile-based ML Apps
- What is a Rest API
- Deploy ML API on Flask Server
- Using ML API from Mobile Apps
- Deploy ML API on Azure Service
- Deploy ML API on PythonAnyWhere Service

Web API on PythonAnywhere

pythonanywhere
by ANACONDA.

Create your account

Username:

Email:

Password:

Password (again):

I agree to the [Terms and Conditions](#) and the [Privacy and Cookies Policy](#), and confirm that I am at least 13 years old.

[Register](#)

We promise not to spam or pass your details on to anyone else.

pythonanywhere
by ANACONDA.

Dashboard Consoles Files Web Tasks Databases

Welcome, [vinhnd](#) [Upgrade Account](#)

Dashboard

CPU Usage: 5% used – 5.82s of 100s. Resets in 24 minutes [More Info](#)

File storage: 10% full – 49.2 MB of your 512.0 MB quota [More Info](#)

Recent Consoles [+ 5 -](#)

Bash console 32772403

New console: [\\$ Bash](#) [>>> Python](#) [More...](#)

Recent Files [+ 5 -](#)

/home/vinhnd/mysite/flask_app.py

/home/vinhnd/mysite/requirements.txt

+ Open another file [Browse files](#)

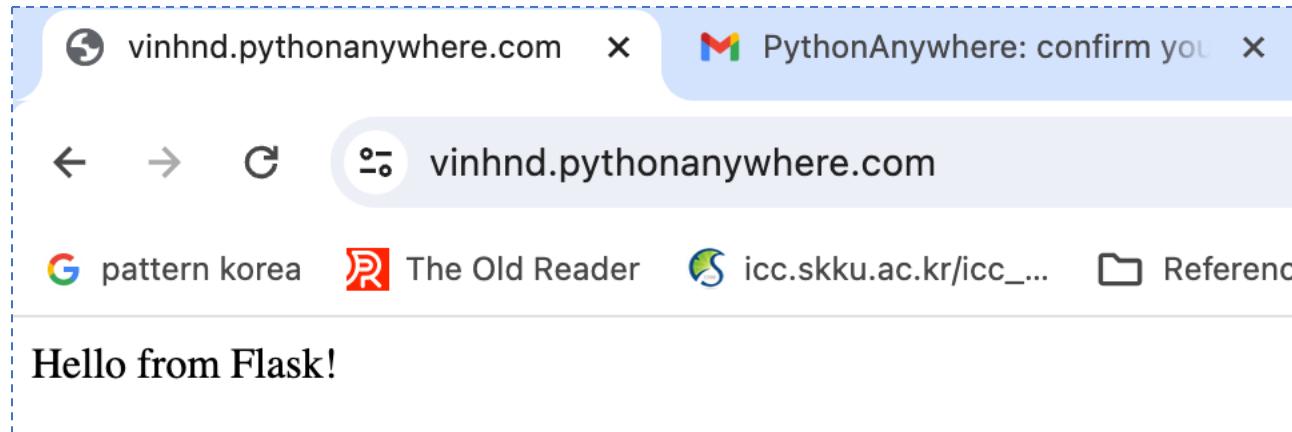
Recent Notebooks [+ 5 -](#)

Your account does not support Jupyter Notebooks. [Upgrade your account](#) to get access!

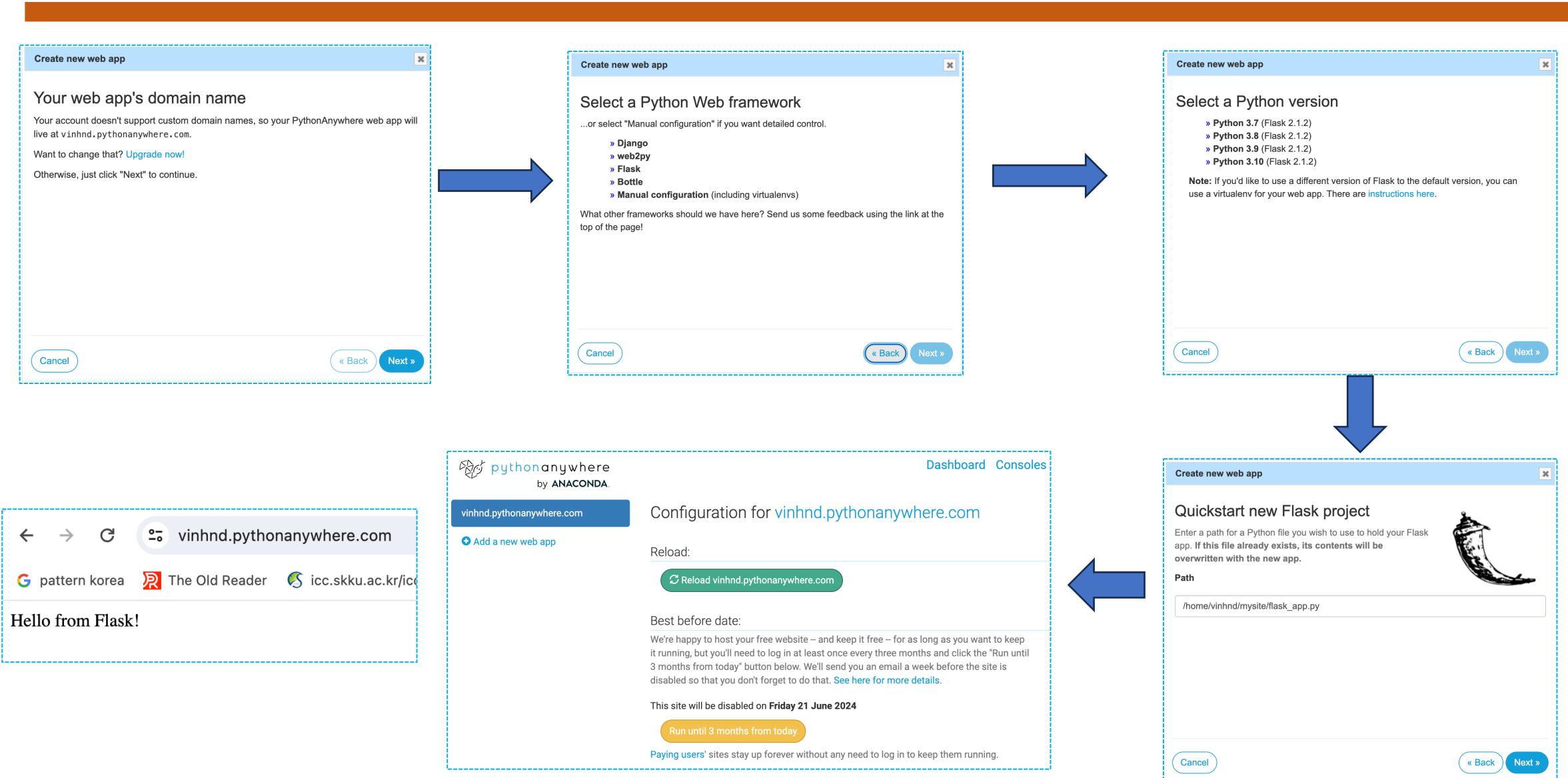
All Web apps

You don't have any web apps.

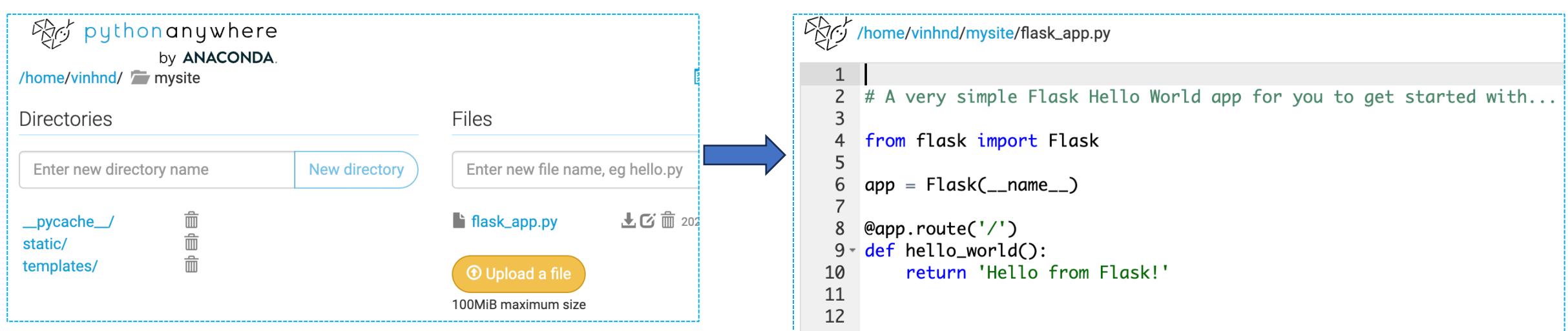
[Open Web tab](#)



Web API on PythonAnyWhere



Web API on PythonAnywhere



Web API on PythonAnyWhere

Start a new console:

Python: [3.10](#) / [3.9](#) / [3.8](#) / [3.7](#) / [3.6](#) IPython: [3.10](#) / [3.9](#) / [3.8](#) / [3.7](#) / [3.6](#) PyPy: [2](#) / [3](#)
Other: [Bash](#) | [MySQL](#)
Custom: [!\[\]\(1ee18ed9aaec3f83a22314fc6c4e909b_img.jpg\)](#)

Your consoles:

Python3.8 console 32772295 ✘

Files

Enter new file name, eg hello.py

flask_app.py

 Upload a file

100MiB maximum size

```
07:58 ~ $ mkvirtualenv flaskenv --python=python3.8
created virtual environment CPython3.8.10.final.0-64 in 11349ms
  creator CPython3Posix(dest=/home/vinhnd/.virtualenvs/flaskenv, clear=False, no_vcs_ignore=False,
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app
    added seed packages: pip==22.1.2, setuptools==62.6.0, wheel==0.37.1
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,P
virtualenvwrapper.user_scripts creating /home/vinhnd/.virtualenvs/flaskenv/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/vinhnd/.virtualenvs/flaskenv/bin/postdeactivate
virtualenvwrapper.user_scripts creating /home/vinhnd/.virtualenvs/flaskenv/bin/preactivate
virtualenvwrapper.user_scripts creating /home/vinhnd/.virtualenvs/flaskenv/bin/postactivate
virtualenvwrapper.user_scripts creating /home/vinhnd/.virtualenvs/flaskenv/bin/get_env_details
```



/home/vinhnd/mysite/flask_app.py (unsaved changes)

Keyboard shortcuts

```
1 import os
2
3 from flask import Flask, redirect, render_template, request,
4                 send_from_directory, url_for)
5
6 app = Flask(__name__)
7
8
9 @app.route('/')
10 def index():
11     print('Request for index page received')
12     return render_template('index.html')
13
14 @app.route('/favicon.ico')
15 def favicon():
16     return send_from_directory(os.path.join(app.root_path, 'static'),
17                               'favicon.ico', mimetype='image/vnd.microsoft.icon')
18
19 @app.route('/hello', methods=['POST'])
20 def hello():
21     name = request.form.get('name')
22
23     if name:
24         print('Request for hello page received with name=%s' % name)
25         return render_template('hello.html', name=name)
26     else:
27         print('Request for hello page received with no name or blank name -- redirecting')
28         return redirect(url_for('index'))
29
30
31 if __name__ == '__main__':
32     app.run()
33
```

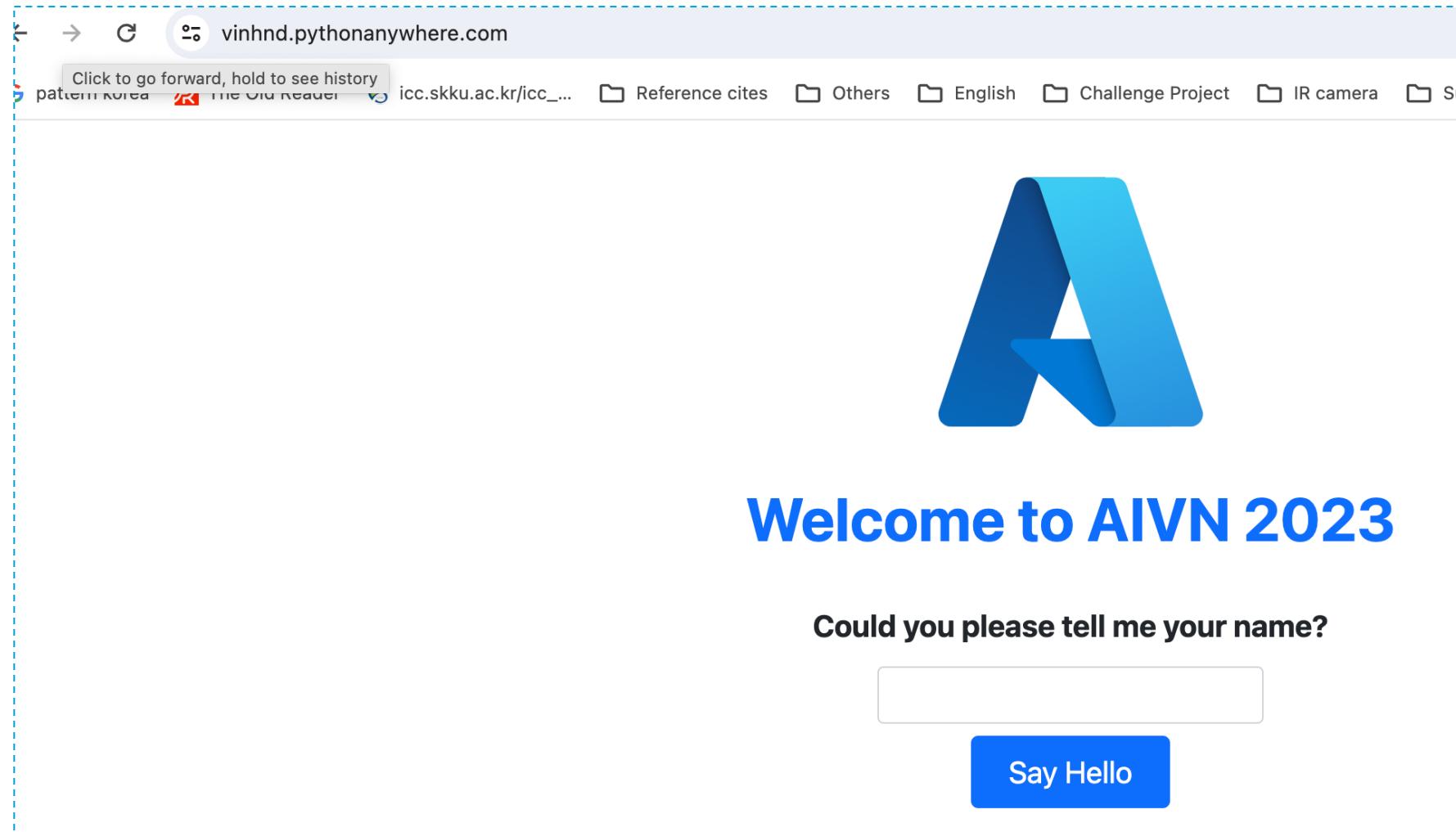
Web API on PythonAnywhere

The screenshot shows the PythonAnywhere dashboard interface. On the left, there's a sidebar with a logo, the text "pythonanywhere by ANACONDA.", and a path "/home/vinhnd/mysite/templates". Below this is a "Directories" section with a text input field "Enter new directory name" and a "New directory" button. To its right is a "Files" section with a text input field "Enter new file name, eg hello.py". It lists two files: "hello.html" and "index.html", each with download, edit, and delete icons. Below these is a "Upload a file" button and the text "100MiB maximum size". On the right side, there's a "Dashb" button and an "Open Bash console" link. A large dashed blue box encloses the main file management area. To the right of this box is a "templates" folder containing "hello.html" and "index.html". Below these are two code editors: one for "index.html" and one for "hello.html". The "index.html" editor shows basic HTML with Bootstrap styling. The "hello.html" editor shows more complex code including imports and a main content block.

```
<!DOCTYPE html>
<head>
    <title>Hello Azure - Python Quickstart</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='bootstrap.css') }}">
    <link rel="shortcut icon" href="{{ url_for('static', filename='icon.ico') }}"/>
</head>
<html>
    <body>
        <main>
            <div class="px-4 py-3 my-2 text-center">
                
                <!-- 
                <h1 class="display-6 fw-bold">Hello {{name}}</h1>
            </div>
        </main>
    </body>
</html>
```

```
<!DOCTYPE html>
<head>
    <title>Hello Azure - Python Quickstart</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='bootstrap.css') }}">
    <link rel="shortcut icon" href="{{ url_for('static', filename='icon.ico') }}"/>
</head>
<html>
    <body>
        <main>
            <div class="px-4 py-3 my-2 text-center">
                
                <!-- 
                <h1 class="display-6 fw-bold">Hello {{name}}</h1>
            </div>
        </main>
    </body>
</html>
```

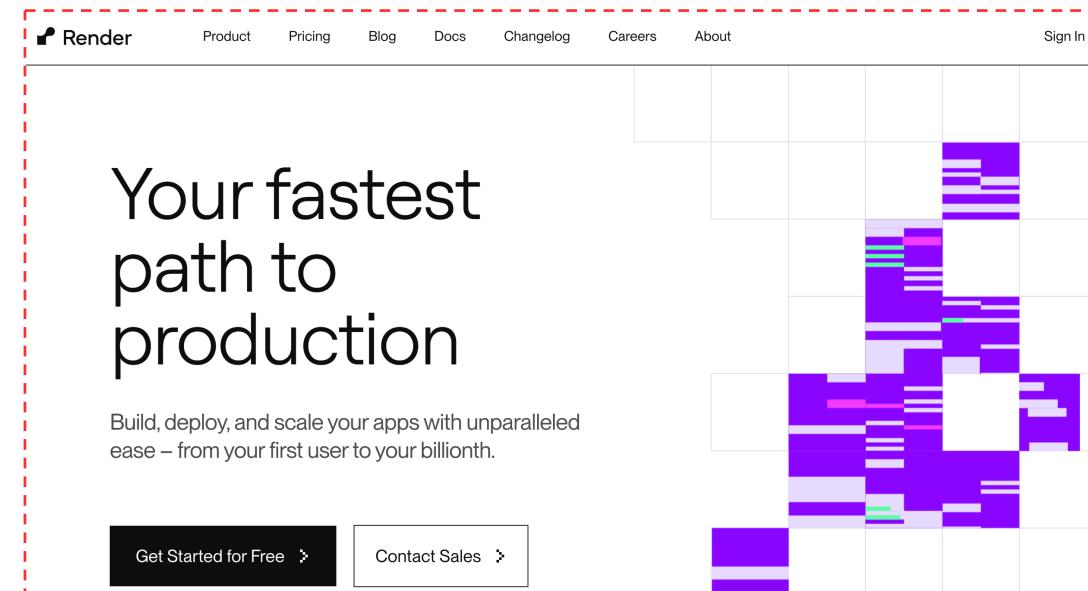
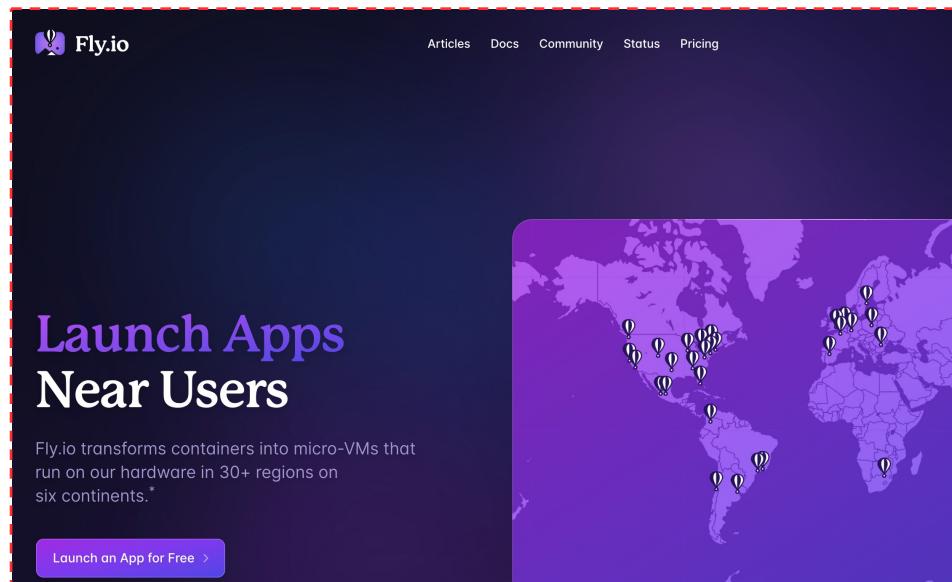
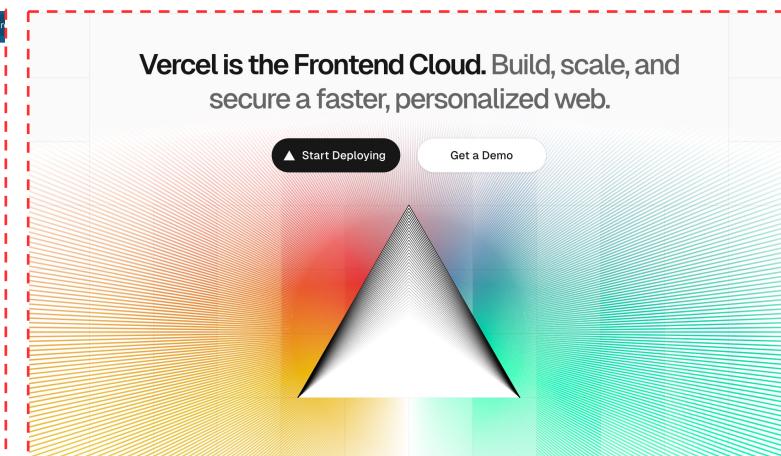
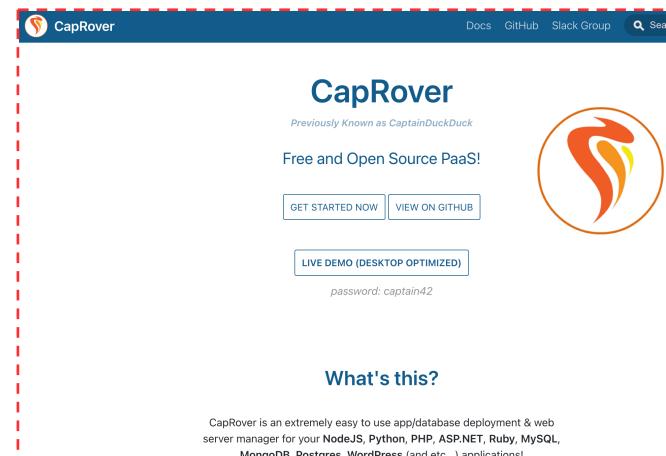
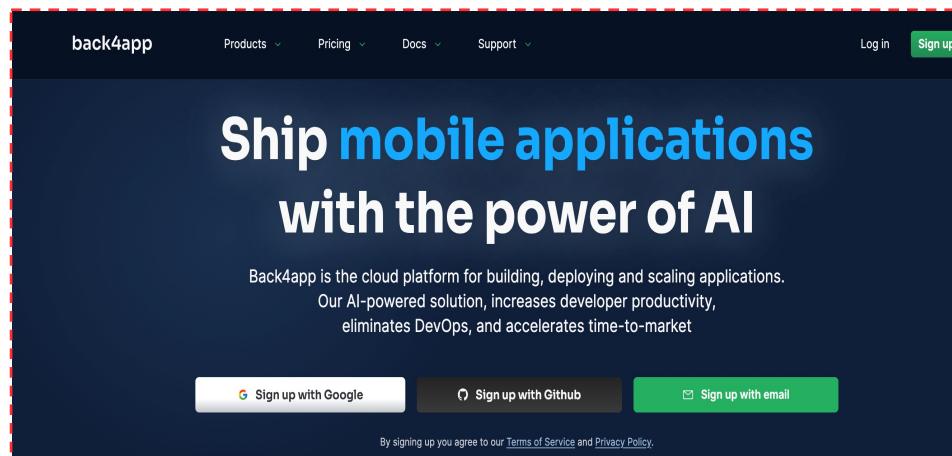
Web API on PythonAnyWhere



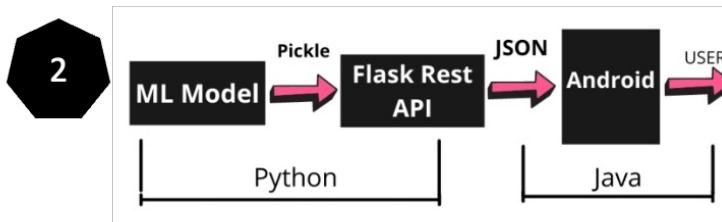
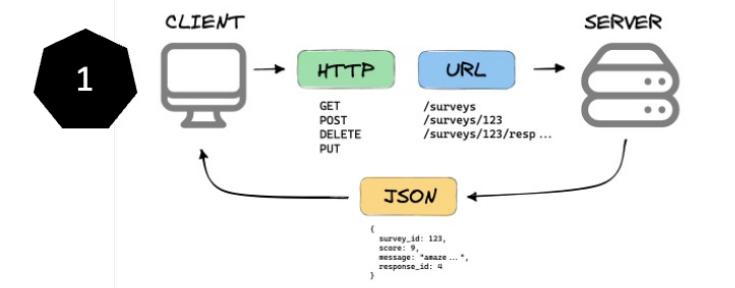
Web API on PythonAnywhere

Hacker	\$5/month	Web dev	\$12/month	Startup	\$99/month	Custom	\$5 to \$500/month
Run your Python code in the cloud from one web app and the console		If you want to host small Python-based websites for you or for your clients		Start a business and don't worry about having to scale to handle traffic spikes		Want a combination that's not on the list? Create your own! All custom plans have:	
A Python IDE in your browser with unlimited Python/bash consoles		A Python IDE in your browser with unlimited Python/bash consoles		A Python IDE in your browser with unlimited Python/bash consoles		A Python IDE in your browser with unlimited Python/bash consoles	
One web app on a custom domain or <i>your-username</i> . pythonanywhere.com		Up to 2 web apps on custom domains or <i>your-username</i> . pythonanywhere.com		Up to 3 web apps on custom domains or <i>your-username</i> . pythonanywhere.com		Up to 20 web apps, on custom domains or <i>your-username</i> . pythonanywhere.com	
Enough power to run a typical 100,000 hit/day website. (more info)		Enough power to run a typical 150,000 hit/day website on each web app. (more info)		Enough power to run a typical 1,000,000 hit/day website on each web app. (more info)		As many web workers as you need to scale your site's capacity. (more info)	
2,000 CPU-seconds per day for consoles, scheduled tasks and always-on tasks (more info)		4,000 CPU-seconds per day for consoles, scheduled tasks and always-on tasks (more info)		10,000 CPU-seconds per day for consoles, scheduled tasks and always-on tasks (more info)		Up to 100,000 CPU-seconds per day for consoles, scheduled tasks and always-on tasks (more info)	
IPython/Jupyter notebook support		IPython/Jupyter notebook support		IPython/Jupyter notebook support		IPython/Jupyter notebook support	
1GB disk space		5GB disk space		50GB disk space		As much disk space as you choose	
Switch Now		Switch Now		Switch Now		Switch Now	

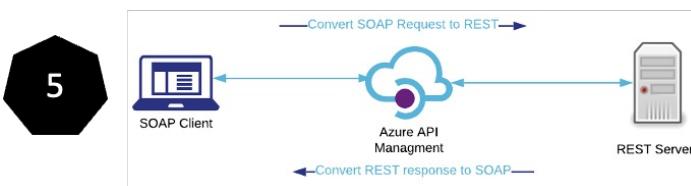
Other Free Hosting Services



Summary



- How to build a simple ML server
- How to connect ML API from mobile to local server
- How to connect ML API from mobile to Azure server
- How to connect ML API from PythonAnywhere Server



Thank
you



