**api**

API های RESTful **:Representational State Transfer (REST)** یک سبک معماری برای ساخت API است. API های RESTful از متدهای HTTP (GET، POST، PUT، DELETE) برای انجام عملیات روی منابع استفاده می‌کنند. آن‌ها به دلیل سادگی، مقیاس‌پذیری و سازگاری با استانداردهای وب بسیار مورداستفاده قرار می‌گیرند.

**کاربردهای API در پایتون**

در توسعه وب مدرن، چندین چارچوب و فناوری وجود دارد که توسعه‌دهندگان را قادر می‌سازد API های کارآمد و قوی بسازند. FastAPI و Flask که هر دو فریمورک‌های پرکاربرد در پایتون هستند، راه‌های یکپارچه‌ای را برای ایجاد API های REST ارائه می‌دهند. FastAPI که به دلیل عملکرد بالای خود شناخته می‌شود، سطح جدیدی از سرعت و مقیاس‌پذیری را برای توسعه API به ارمغان می‌آورد. از سوی دیگر، Flask، یک چارچوب سبک‌وزن، انعطاف‌پذیری و سادگی را برای ساخت سرویس‌های RESTful فراهم می‌کند. برای مثال، می‌توان با استفاده از Python Flask، یک مثال REST API ایجاد کرد و از قابلیت‌های مسیریابی و رسیدگی بصری آن استفاده کرد.

**کار با API در پایتون**

به‌صورت کلی کار با API در پایتون شامل سه مرحله کلی زیر است:

1. ایجاد درخواست‌های HTTP: پایتون چندین کتابخانه مانند Requests، urllib و httplib2 را برای ایجاد درخواست‌های HTTP به API ها ارائه می‌دهد. این کتابخانه‌ها فرآیند ارسال درخواست و دریافت پاسخ از نقاط پایانی API را ساده می‌کنند.
2. تجزیه پاسخ‌های JSON: API ها اغلب داده‌ها را در قالب JSON برمی‌گردانند. کتابخانه JSON داخلی پایتون به شما امکان می‌دهد به‌راحتی داده‌ها را از پاسخ‌های JSON تجزیه و استخراج کرده و به شما امکان می‌دهد با اطلاعات بازیابی شده به‌طور یکپارچه کارکنید.
3. مدیریت احراز هویت: بسیاری از API ها برای اطمینان از دسترسی ایمن نیاز به احراز هویت دارند. کتابخانه‌های پایتون مکانیسم‌هایی را برای مدیریت روش‌های احراز هویت مختلف، مانند کلیدهای API، OAuth و احراز هویت مبتنی بر توکن فراهم می‌کنند.

**کتابخانه‌ های پایتون برای کار با API**

کتابخانه‌های زیادی برای کار با API با پایتون وجود دارد که از مهم‌ترین این کتابخانه‌ها می‌توان موارد زیر را نام برد:

* Requests**:Requests** نوعی کتابخانه HTTP محبوب در پایتون است که انجام درخواست‌های HTTP را ساده می‌کند. این یک API کاربرپسند برای ارسال درخواست‌های GET، POST، PUT، DELETE و مدیریت داده‌های پاسخ ارائه خواهد کرد.
* urllib **:urllib** نوعی کتابخانه داخلی پایتون است که کلاس‌ها و توابعی را برای مدیریت URL ها، ایجاد درخواست‌های HTTP و پردازش پاسخ‌ها ارائه می‌دهد. این رابط کاربری سطح پایین‌تری را در مقایسه با Requests ارائه می‌دهد، اما همچنان به‌طور گسترده برای یکپارچه سازی API استفاده می‌شود.
* httplib2 **:httplib2** کتابخانه دیگری است که امکان ایجاد درخواست‌های HTTP در پایتون را می‌دهد. این برنامه از ویژگی‌هایی مانند ذخیره‌سازی، احراز هویت و مدیریت تغییر مسیرها پشتیبانی می‌کند که آن را به یک انتخاب قدرتمند برای یکپارچه سازی API تبدیل می‌کند.

**نمونه های عملی api نویسی در پایتون**

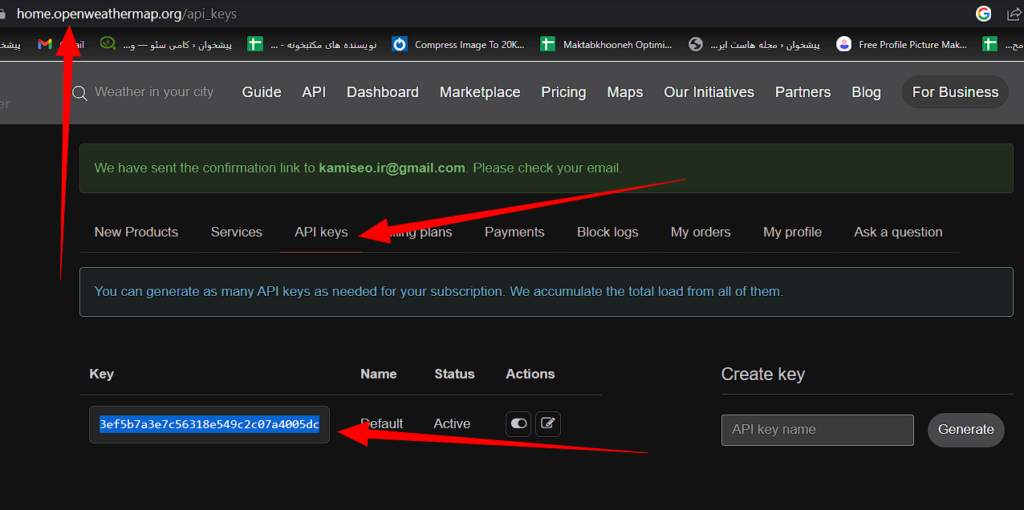
از مهم‌ترین نمونه های عملی استفاده و نوشتن api در پایتون می‌توان به موارد زیر اشاره کرد:

* واکشی داده‌های آب‌وهوا
* دسترسی به API های رسانه‌های اجتماعی
* ادغام با خدمات وب
* و بسیاری از موارد دیگر

در این مطلب ما، نحوه استفاده از api در پایتون برای تعامل با API آب‌وهوا و بازیابی اطلاعات آب‌وهوای فعلی بر اساس مکان را نشان خواهیم داد. ما از OpenWeatherMap API استفاده خواهیم کرد که داده‌های آب‌وهوا را برای مکان‌های مختلف در سراسر جهان ارائه می‌دهد. برای شروع، باید مراحل زیر را دنبال کنید.

**مرحله 1: ثبت‌نام و دریافت کلید API:**

برای دسترسی به OpenWeatherMap API، باید برای یک حساب کاربری ثبت‌نام کنید و یک کلید API دریافت کنید. کلید API برای احراز هویت درخواست‌های شما و اطمینان از داشتن مجوز دسترسی به داده‌های آب‌وهوا مورد نیاز است.



**مرحله 2: نصب کتابخانه‌های لازم:**

ما باید کتابخانه requests را نصب کنیم که ایجاد درخواست‌های HTTP در پایتون را ساده می‌کند. برای این هدف باید ترمینال یا خط فرمان خود را باز کرده و دستور زیر را اجرا کنید:

pip install requests

**مرحله 3: ایمپورت کردن کتابخانه‌های مورد نیاز:**

در محیط توسعه‌ یکپارچه یا همان محیط برنامه‌نویسی باید با [دستور import در پایتون](https://maktabkhooneh.org/mag/import-in-python/)، کتابخانه‌های لازم را وارد کنید:

import requests

**مرحله 4: ساخت درخواست API:**

برای واکشی اطلاعات آب‌وهوا، باید URL درخواست API را با پارامترهای مورد نظر بسازید. برای مثال می‌توانید مکان (نام شهر یا مختصات) و واحد دما (سانتی‌گراد یا فارنهایت) را مشخص کنید.

api\_key = "YOUR\_API\_KEY"

location = "tehran" # Change this to the desired location

units = "C" # Change this to "imperial" for Fahrenheit

url = f"http://api.openweathermap.org/data/2.5/weather?q={location}&amp;units={units}&amp;appid={api\_key}"

**مرحله 5: ارسال درخواست API و مدیریت پاسخ:**

تا به اینجا ما از کتابخانه requests برای ارسال درخواست API و بازیابی پاسخ استفاده خواهیم کرد.

response = requests.get(url)

data = response.json()

**مرحله 6: استخراج و نمایش اطلاعات آب‌وهوا:**

داده‌های آب‌وهوا در قالب JSON برگردانده می‌شود. می‌توانید اطلاعات مربوطه را استخراج کرده و نمایش دهید.

if data["cod"] == 200:

temperature = data["main"]["temp"]

weather\_description = data["weather"][0]["description"]

humidity = data["main"]["humidity"]

wind\_speed = data["wind"]["speed"]

print(f"Weather in {location}:")

print(f"Temperature: {temperature}°C")

print(f"Weather description: {weather\_description}")

print(f"Humidity: {humidity}%")ا

print(f"Wind speed: {wind\_speed} m/s")

else:

print(f"Error: {data['message']}")

**مرحله 7: اجرای کد پایتون:**

اسکریپت را با پسوند.py (به عنوان مثال weather.py) ذخیره کرده و آن را اجرا کنید. باید اطلاعات آب‌وهوای مکان مشخص شده را روی کنسول ببینید. تا به اینجا شما با موفقیت آموزش استفاده از api در پایتون آموختید و داده‌های آب‌وهوا را با استفاده از OpenWeatherMap API در پایتون بازیابی کرده‌اید. می‌توانید مکان و سایر پارامترها را برای واکشی داده‌های آب‌وهوا برای مکان‌های مختلف تغییر دهید یا خروجی را مطابق با نیاز خود سفارشی کنید. برای مثال ما وضعی آب و هوا را برای شهر تهران به صورت زیر دریافت کردیم:

**بهترین روش ها برای کار با API ها در پایتون**

برای اینکه استفاده از API در پایتون به بهترین شکل ممکن انجام بگیرد نیاز است که روش‌های استفاده از آن را به نحو احسن به کار بگیرید که در این رابطه توجه به نکات زیر بسیار حائز اهمیت است:

* **خواندن اسناد API:** برای درک نقاط پایانی موجود، پارامترهای درخواست و فرمت‌های پاسخ، با اسناد API آشنا شوید. این به شما کمک می‌کند تا از API به‌طور مؤثر استفاده کرده و از خطاهای غیرضروری جلوگیری کنید.
* **رسیدگی به خطاها و استثناها: API** ها ممکن است پاسخ‌های خطا را برگردانند یا در طول درخواست‌ها با استثناهایی مواجه شوند. مدیریت صحیح خطا را برای رسیدگی به این موقعیت‌ها و ارائه بازخورد معنی‌دار به کاربران پیاده‌سازی کنید.
* **محدودیت نرخ پیاده‌سازی: API** ها اغلب برای جلوگیری از سوءاستفاده، محدودیت‌هایی را اعمال می‌کنند. با اجرای مکانیسم‌های محدودکننده نرخ در کد خود، به این محدودیت‌ها پایبند باشید تا از مسدود شدن یا بلاک شدن توسط ارائه‌دهنده API جلوگیری کنید.
* **تماس‌های امن API:** اگر یک API نیاز به احراز هویت دارد، اطلاعات حساس را به‌طور ایمن مدیریت کنید. اعتبارنامه‌ها را به روشی ایمن ذخیره کرده و از روش‌های رمزگذاری یا احراز هویت مبتنی بر توکن برای محافظت از داده‌ها در طول تماس‌های API استفاده کنید.

**سخن پایانی**

در این مقاله به اصول آموزش کار با api در پایتون، از جمله تعریف، انواع، پیاده‌سازی و بهترین روش‌ها پرداختیم. ما نحوه کار با API ها را با استفاده از کتابخانه‌های پایتون مانند Requests ،urllib و httplib2 بررسی کردیم. علاوه بر این، نمونه‌های عملی از یکپارچه‌سازی API در پایتون، از جمله واکشی داده‌های آب‌وهوا را نیز به‌صورت عملی انجام دادیم. با پیروی از بهترین روش‌های مورد بحث، می‌توانید یکپارچه‌سازی API خود را بهینه کرده و از تجربه‌ای روان و ایمن اطمینان حاصل کنید.