

The God We Trust



2nd Assignment Report

<Mitra Dehghan> <98222038>

Introduction:

هدف کلی این پروژه ساخت یک برنامه جاوا می باشد که به یک API آب و هوا وصل شده و اطلاعات مد نظر را میگیرد و زمانی که اسم شهر را به عنوان ورودی به آن میدهیم وضعیت آب و هوایی آن را به ما بر میگردداند که البته در این پروژه تمرکز روی دما و رطوبت آن شهر است .

یکی از ابزارهای در دست برای پیش برد این پروژه `Json library` بود که باید آن را در ابتدای ماجرا در فایل `build.gradle` اضافه میکردیم تا بتوانیم برای ادامه کار از آن استفاده کنیم .

Design and Implementation:

در وهله اول برای بدست آوردن اطلاعات مورد نیاز ، در سایت `www.weatherapi.co` عضو شده و اول `API Key` را به طور اختصاصی ساختیم و سپس از قسمت `API Explorer` با کامل کردن جزئیات خواسته شده و قرار دادن `API` ای که در مرحله قبل ساختیم ، اطلاعات مورد نظر به شکل زیر به ما داده شد.

```
http://api.weatherapi.com/v1/current.json?key=0859916d42244043b0862754232802&q=Tehran&aqi=no
```

200

```
{
  "Transfer-Encoding": "chunked",
  "Connection": "keep-alive",
  "Vary": "Accept-Encoding",
  "CDN-PullZone": "93447",
  "CDN-Uid": "8fa3a04a-75d9-4707-8056-b7b33c8ac7fe",
  "CDN-RequestCountryCode": "GB",
  "CDN-ProxyVer": "1.03",
  "CDN-RequestPullSuccess": "True",
  "CDN-RequestPullCode": "200",
  "CDN-CachedAt": "03/01/2023 19:04:46",
  "CDN-EdgeStorageId": "1074",
  "CDN-Status": "200",
```

```

"CDN-RequestId": "aa42b08c32df7fa720268f6f2108a8ad",
"CDN-Cache": "EXPIRED",
"Cache-Control": "public, max-age=180",
"Content-Type": "application/json",
"Date": "Wed, 01 Mar 2023 19:04:46 GMT",
"Server": "BunnyCDN-FR1-1072"
}

{
  "location": {
    "name": "Tehran",
    "region": "Tehran",
    "country": "Iran",
    "lat": 35.73,
    "lon": 51.33,
    "tz_id": "Asia/Tehran",
    "localtime_epoch": 1677697486,
    "localtime": "2023-03-01 22:34"
  },
  "current": {
    "last_updated_epoch": 1677697200,
    "last_updated": "2023-03-01 22:30",
    "temp_c": 10.0,
    "temp_f": 50.0,
    "is_day": 0,
    "condition": {
      "text": "Clear",
      "icon": "//cdn.weatherapi.com/weather/64x64/night/113.png",
      "code": 1000
    },
    "wind_mph": 4.3,
    "wind_kph": 6.8,
    "wind_degree": 180,
    "wind_dir": "S",
    "pressure_mb": 1019.0,
    "pressure_in": 30.1,
    "precip_mm": 0.0,
    "precip_in": 0.0,
    "humidity": 29,
    "cloud": 0,
    "feelslike_c": 9.2,
    "feelslike_f": 48.5,
    "vis_km": 10.0,
    "vis_miles": 6.0,
    "uv": 1.0,
    "gust_mph": 8.5,
    "gust_kph": 13.7
  }
}

```

که در کل سه قسمت آن برای ما در اینجا کارایی دارد :

"temp_c"/ "humidity" / "current"

(از اطلاعات بالا میتوان در مرحله تست برای چک کردن جواب ها استفاده کرد)
بعد از این عملیات ، مقدار **API Key** را به عنوان ثابت خود در کد قرار دادیم .
در گام بعدی نیاز به تعریف دو تابع **getTemperature** و **getHumidity** داشتیم که به ترتیب یکی مربوط به گرفتن مقدار دمای آن شهر و دیگری برای رطوبت آن بود .

عملکرد هردو این توابع مشابه هم بود و برای ساخت و تکمیل هردو آنها ابتدا یک **JSONObject** ساختیم که آرگومان هردوی آن ها از جنس **String** تعریف شد .
یک متغیر **answer** تعریف کردیم که در اصل جواب نهایی ای می باشد که قرار است این تابع به ما برگرداند و آن هم همان مقداریست که ما به ازای **current** و **temp_c** (شهر مورد نظر و میزان دمای آن با توجه به **API Key** وارد شده در همان سایتی که اول روی آن اکانت ساختیم) و به طور مشابه **current** و **humidity** برای مجددا شهر مورد نظر و میزان رطوبت آن .
که برای گرفتن و ذخیره سازی این سه مورد به ترتیب از توابع **getJSONObject** و **getInt** و **getDouble** استفاده کردیم .

بعد از تعریف دو تابع مذکور به سراغ تابع **main** میرویم و الان فقط کافیست در ابتدا از کاربر بخواهیم که نام شهر مورد نظرش را وارد کند ، در مرحله بعد رشته ای با عنوان **json_Data** ساخته میشود که نام شهر را به تابع **getWeatherData** میبرد و در آن چک میشود که اگر این نام با شهری که در آن سایت اطلاعاتش موجوده تطابق داده باشد آن پایگاه داده به نوبعی در اختیار ما قرار میگیرد و از دو عنصر دما و رطوبت آن میتوان استفاده کرد.

ادامه کار هم به این شکل است که در همان تابع **main** ، دو دستور برای نمایش میزان دما و رطوبت نوشته شده که در آنها ابتدا توابع تعریف شده صدا زده میشود و مقدار مدنظر برمیگردد و در صفحه به نمایش گذاشته میشود.

Testing and Evaluation:

وقتی که برنامه را اجرا میکنیم و از ما خواسته میشود که نام شهر مورد نظر را وارد کنیم ، با وارد کردن Tehran ، اعدادی را به ترتیب برای میزان دمای شهر تهران و میزان رطوبت آن در ساعت و تاریخی که این API Key ساخته شد نمایان میشود . همانطور که بالاتر اشاره شد میتوان برای چک کردن درستی اعداد نمایش داده شده به همان سایت اول رجوع کرد .

Conclusion:

در کنار اینکه این پروژه میتواند در عین سادگی بسیار کاربردی باشد اما یکی از ضعف های برنامه که بیشتر از هرچیز در نگاه من مطرح بود در رابطه با سرعت آن میباشد چرا که مدت زیادی طول میکشد تا اجرا شود .