MONASH University

FIT2081 Mobile Application Development

WEEK 6

Dr. Lim Chern Hong

Semester 1, 2023

Monash University Malaysia

Announcement for Week 6

- Please complete your pre-reading quiz and submit by Monday 4pm. You can find the pre-reading quiz link at moodle “assessment” section.

- You will have to complete and submit the workshop quiz which will make available after the forum by Wednesday 11.55pm.

- Please complete your lab tasks before joining your lab session. Your lab solution must be submitted to moodle by Friday 11.55pm.

Learning Outcomes for Week 6

- Examine the proper use of the RecyclerView and CardViews.
- Understand the correct methods for creating RecyclerViews.
- Understand when Fragments should be used and how to manage them.
- Examine how to manage resources from a webservice.

Activities and Checklist for week 6

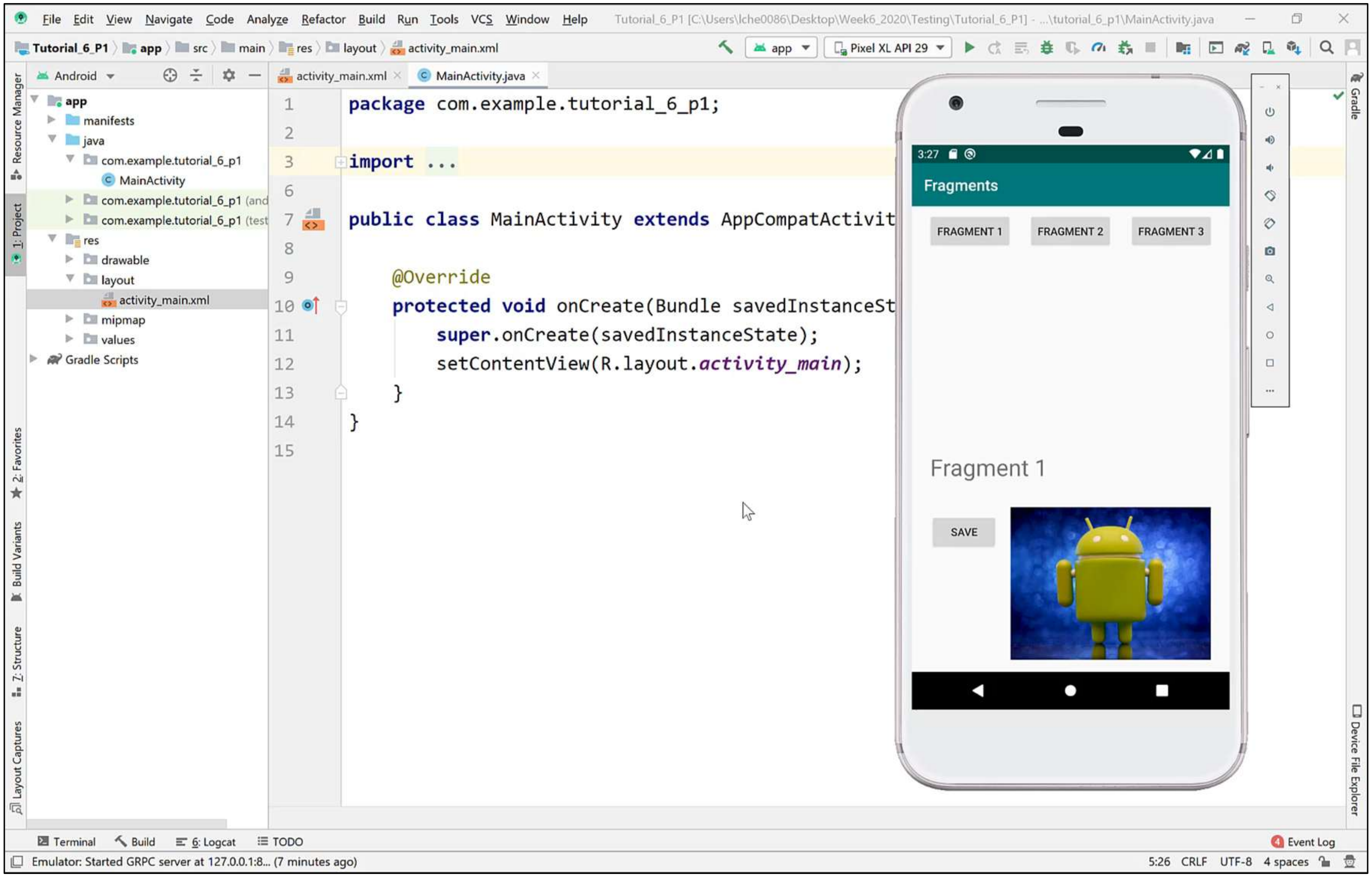
Activity	Notes	Checked?
Study the slide “FIT2081_Week6_Malaysia” & All the reading material in the moodle	Useful to complete your lab tasks.	
Complete the pre-reading quiz	Access it from the “assessment section” in moodle. Submit by Monday 4pm.	
Attend Forum	Online, for topics wrap-up.	
Complete Workshop quiz	Workshop quiz questions will be uploaded after the Forum on Monday. Submit by Wednesday 11.55pm.	
Complete lab task	Please refer to the complete section in week 6 moodle	
Attend tutorial	OPTIONAL – if you have issue regarding the lab tasks	
Attend Lab	COMPULSORY – You have to complete the lab tasks before coming to the lab. Submit your lab tasks (including the extra task) on Friday 11.55pm	

Tutorial time!

GIF retrieved from https://gfycat.com

Video 1: Fragments (sub activities)

1) Please play the video (36 minutes 02 seconds)



```
package com.example.tutorial_6_p1;

import ...

public class MainActivity extends AppCompatActivit

    @Override
    protected void onCreate(Bundle savedInstanceSt
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

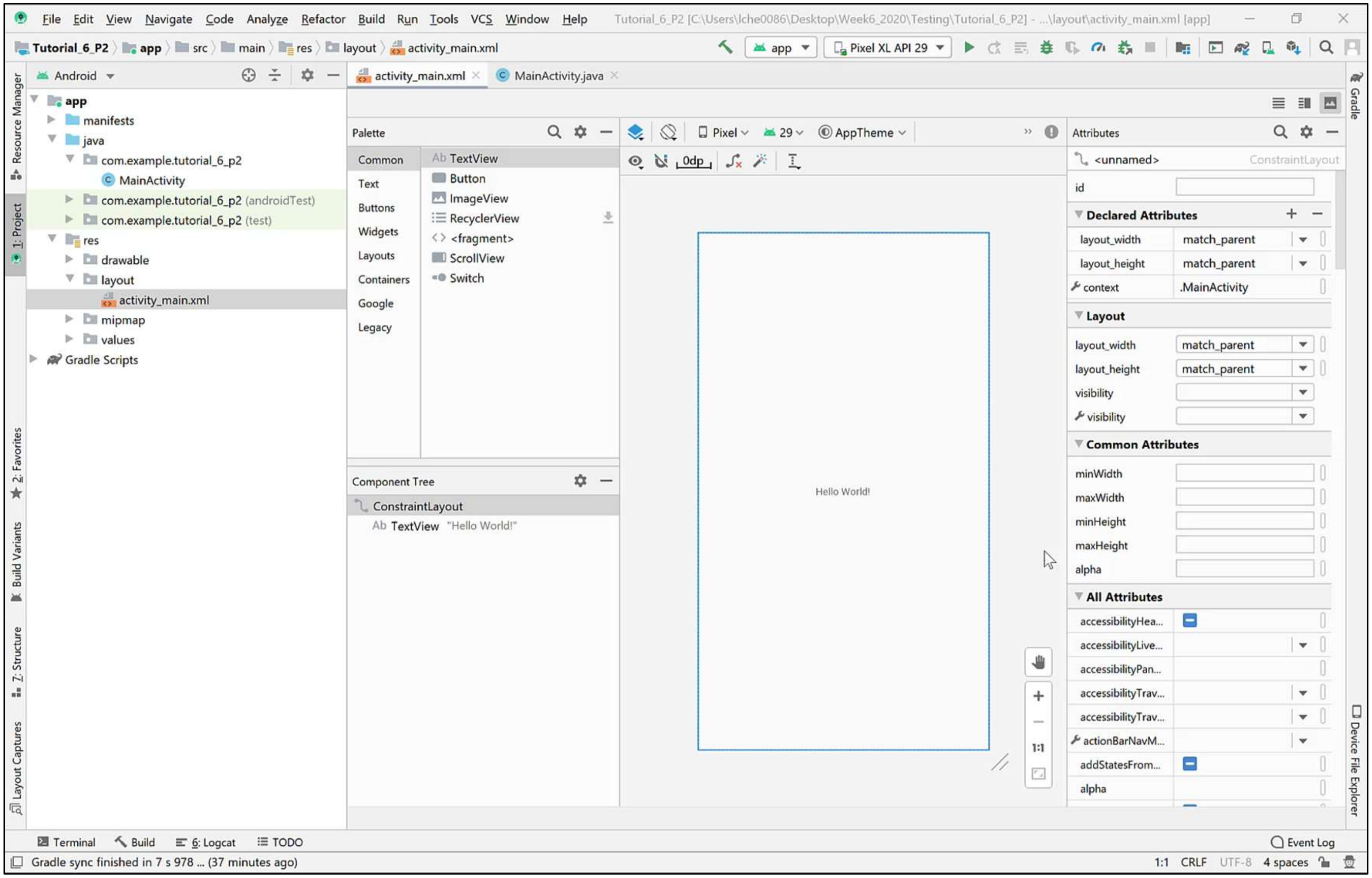
Emulator: Started GRPC server at 127.0.0.1:8... (7 minutes ago)

Note:
26:53:00 – You can do it in onActivityCreated lifecycle callback as well which comes after onViewCreated lifecycle.

You may refer to the complete android lifecycle here: https://i.stack.imgur.com/qyU1H.png

Video 2: CardView and RecyclerVeiw

1) Please play the video (34 minutes 09 Seconds)



Declared Attributes

layout_width	match_parent
layout_height	match_parent
context	.MainActivity

Layout

layout_width	match_parent
layout_height	match_parent
visibility	
visibility	

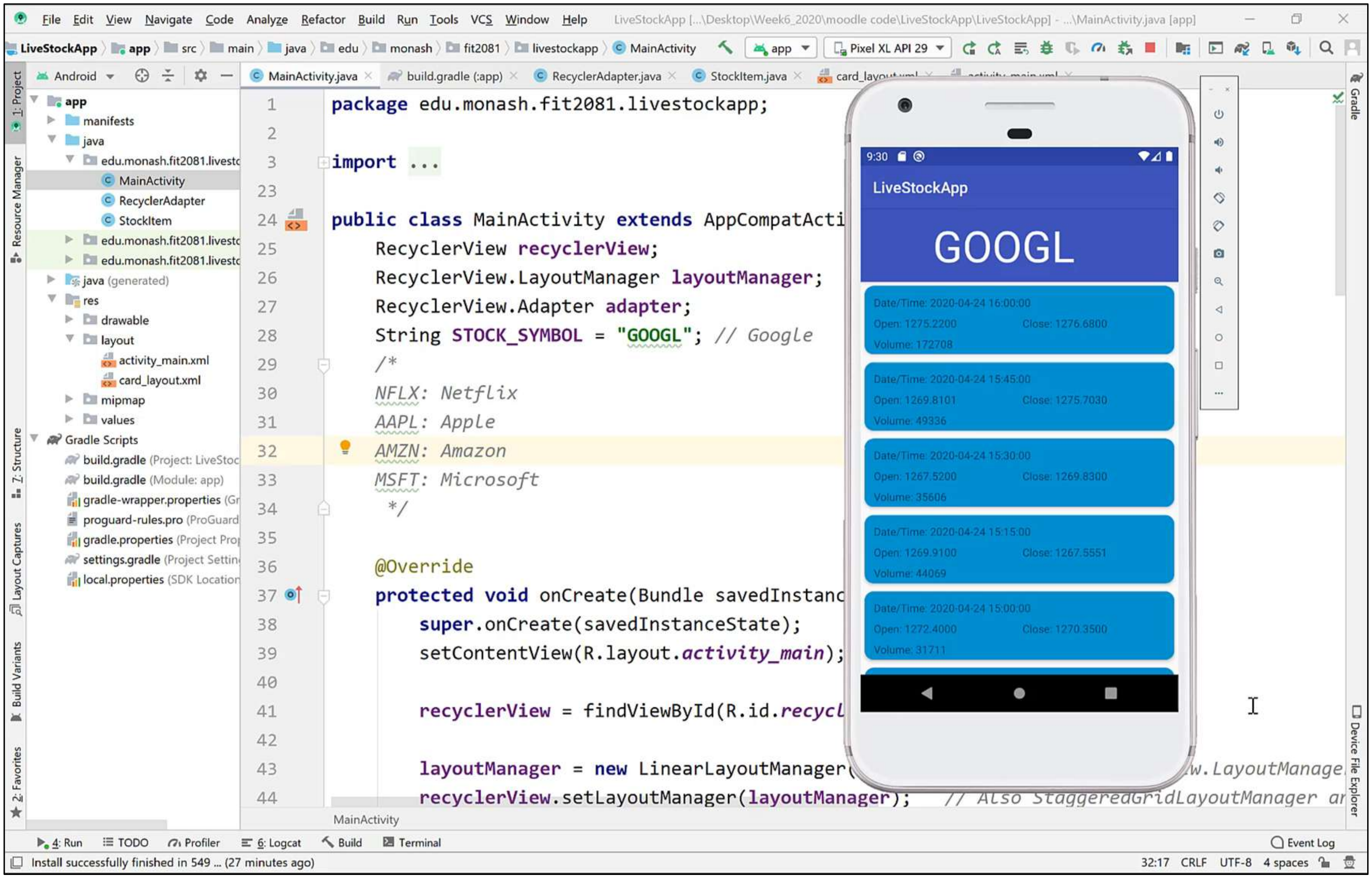
Common Attributes: minWidth, maxWidth, minHeight, maxHeight, alpha

All Attributes: accessibilityHea..., accessibilityLive..., accessibilityPan..., accessibilityTrav..., accessibilityTrav..., actionBarNavM..., addStatesFrom..., alpha

Gradle sync finished in 7 s 978 ... (37 minutes ago)

Video 3: Extract information from Webservice

1) Please play the video (13 minutes 52 Seconds)


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
LiveStockApp [...\Desktop\Week6_2020\moodle code\LiveStockApp\LiveStockApp] - ...\MainActivity.java [app]
LiveStockApp 〉 app 〉 src 〉 main 〉 java 〉 edu 〉 monash 〉 fit2081 〉 livestockapp 〉 MainActivity
app
Pixel XL API 29
Android
MainActivity.java
build.gradle (:app)
RecyclerAdapter.java
StockItem.java
card_layout.xml
activity_main.xml
app
manifests
java
edu.monash.fit2081.livestc
MainActivity
RecyclerAdapter
StockItem
edu.monash.fit2081.livestc
edu.monash.fit2081.livestc
java (generated)
res
drawable
layout
activity_main.xml
card_layout.xml
mipmap
values
Gradle Scripts
build.gradle (Project: LiveStoc
build.gradle (Module: app)
gradle-wrapper.properties (Gr
proguard-rules.pro (ProGuard
gradle.properties (Project Pro
settings.gradle (Project Settin
local.properties (SDK Location
1: Project
Resource Manager
7: Structure
Layout Captures
Build Variants
2: Favorites
Gradle
Device File Explorer
MainActivity
4: Run
TODO
Profiler
6: Logcat
Build
Terminal
Event Log
32:17 CRLF UTF-8 4 spaces
9:30
LiveStockApp

```java
package edu.monash.fit2081.livestockapp;

import ...

public class MainActivity extends AppCompatActi
    RecyclerView recyclerView;
    RecyclerView.LayoutManager layoutManager;
    RecyclerView.Adapter adapter;
    String STOCK_SYMBOL = "GOOGL"; // Google
    /*
    NFLX: Netflix
    AAPL: Apple
    AMZN: Amazon
    MSFT: Microsoft
     */

    @Override
    protected void onCreate(Bundle savedInstanc
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        recyclerView = findViewById(R.id.recycl

        layoutManager = new LinearLayoutManager(                w.LayoutManage
        recyclerView.setLayoutManager(layoutManager);   // Also StaggeredGridLayoutManager ar
```

GOOGL

```
Date/Time: 2020-04-24 16:00:00
Open: 1275.2200            Close: 1276.6800
Volume: 172708

Date/Time: 2020-04-24 15:45:00
Open: 1269.8101            Close: 1275.7030
Volume: 49336

Date/Time: 2020-04-24 15:30:00
Open: 1267.5200            Close: 1269.8300
Volume: 35606

Date/Time: 2020-04-24 15:15:00
Open: 1269.9100            Close: 1267.5551
Volume: 44069

Date/Time: 2020-04-24 15:00:00
Open: 1272.4000            Close: 1270.3500
Volume: 31711
```

Install successfully finished in 549 ... (27 minutes ago)

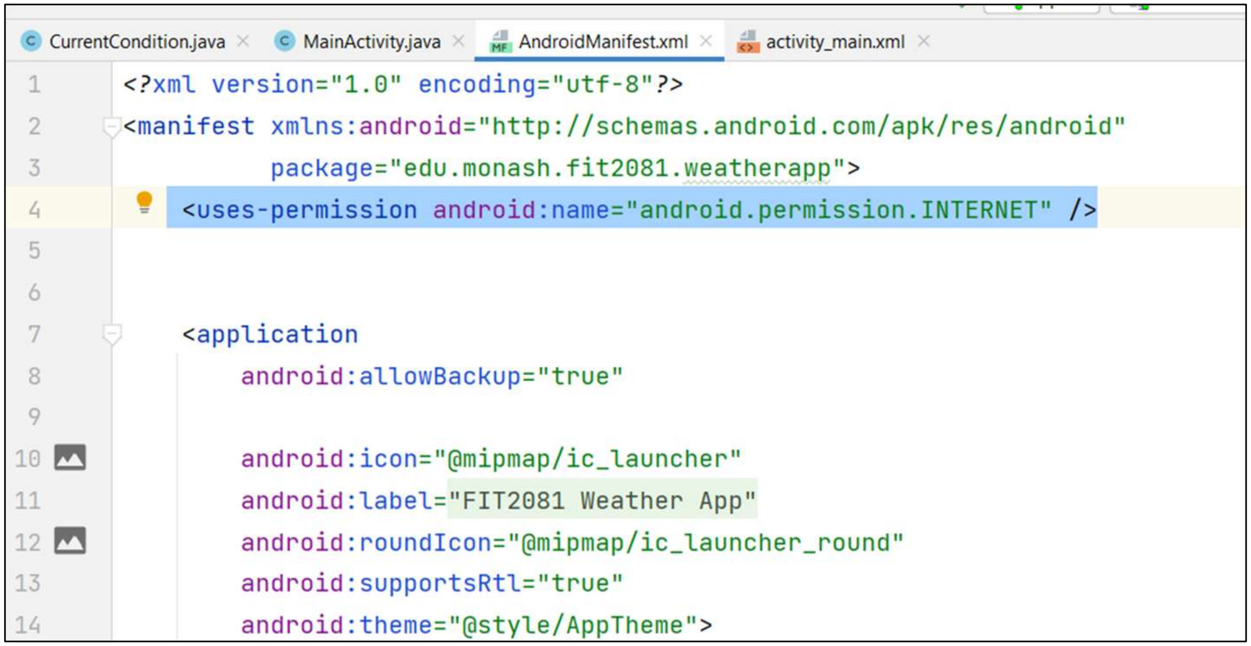
Good reference for lab task2!

Extract information from Webservice

1) Please do not forget to provide permission to your phone for the internet access in your AndroidManifest.xml, else you will not get any response from the webservice:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Example:

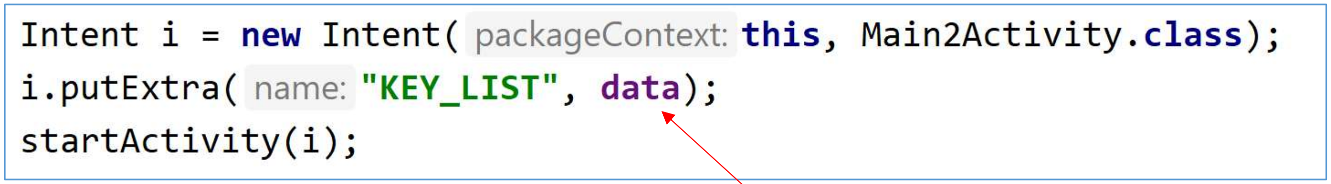


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="edu.monash.fit2081.weatherapp">
    <uses-permission android:name="android.permission.INTERNET" />


    <application
        android:allowBackup="true"

        android:icon="@mipmap/ic_launcher"
        android:label="FIT2081 Weather App"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
```

How to pass Arraylist to another activity?

1) By using Intent (Cannot save persistent data)

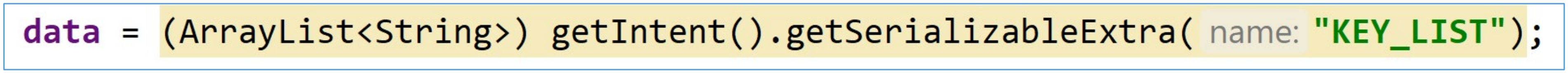
Put it anywhere in your **main activity** that you intent to send to another activity, normally after a button is clicked.



```
Intent i = new Intent( packageContext: this, Main2Activity.class);
i.putExtra( name: "KEY_LIST", data);
startActivity(i);
```

This is your Arraylist

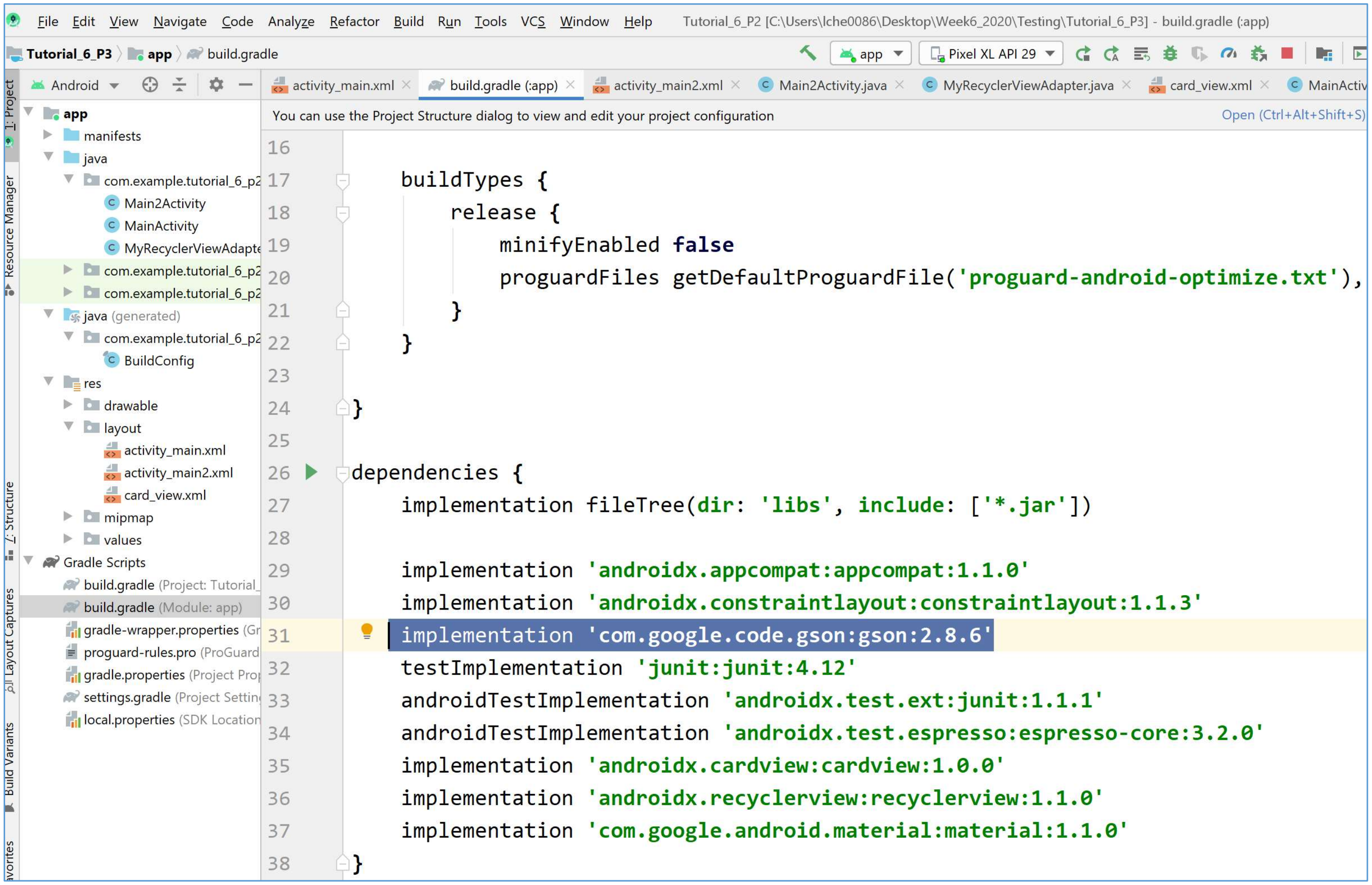
Call this to retrieve your Arraylist in another activity. Then you can get back your array list for other purpose.



```
data = (ArrayList<String>) getIntent().getSerializableExtra( name: "KEY_LIST");
```

How to pass Arraylist to another activity?

2) By using Gson and sharePreference (Preferable, it uses sharePreference which can save data persistently). But first you have to include this dependencies and resync your application.



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Tutorial_6_P2 [C:\Users\lche0086\Desktop\Week6_2020\Testing\Tutorial_6_P3] - build.gradle (:app)
Tutorial_6_P3 › app › build.gradle
app
Pixel XL API 29
Android
activity_main.xml
build.gradle (:app)
activity_main2.xml
Main2Activity.java
MyRecyclerViewAdapter.java
card_view.xml
MainActiv
1: Project
Resource Manager
7: Structure
Layout Captures
Build Variants
avorites
app
manifests
java
com.example.tutorial_6_p2
Main2Activity
MainActivity
MyRecyclerViewAdapte
com.example.tutorial_6_p2
com.example.tutorial_6_p2
java (generated)
com.example.tutorial_6_p2
BuildConfig
res
drawable
layout
activity_main.xml
activity_main2.xml
card_view.xml
mipmap
values
Gradle Scripts
build.gradle (Project: Tutorial_
build.gradle (Module: app)
gradle-wrapper.properties (Gr
proguard-rules.pro (ProGuard
gradle.properties (Project Prop
settings.gradle (Project Settin
local.properties (SDK Location
Open (Ctrl+Alt+Shift+S)

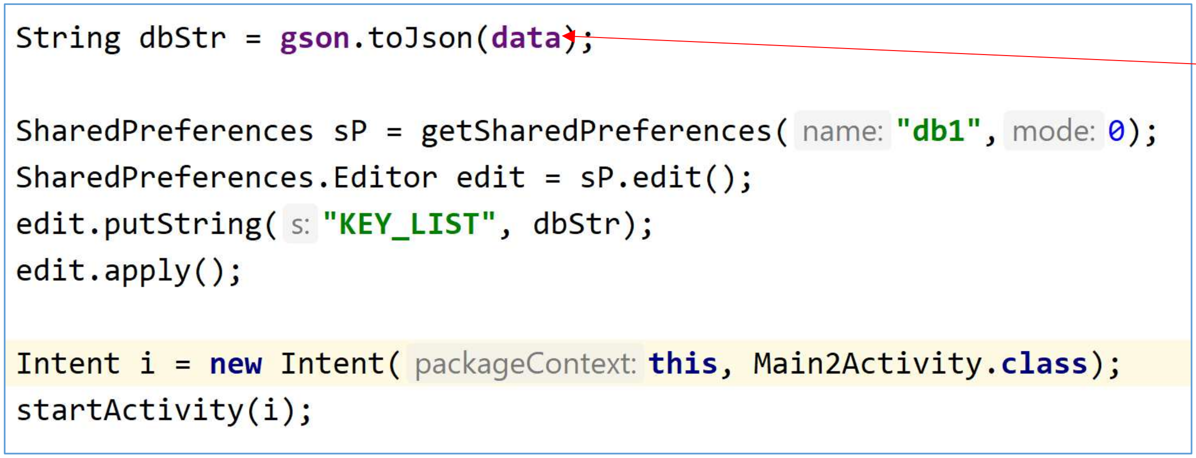
You can use the Project Structure dialog to view and edit your project configuration

```

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
        }
    }

}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'com.google.code.gson:gson:2.8.6'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
    implementation 'androidx.cardview:cardview:1.0.0'
    implementation 'androidx.recyclerview:recyclerview:1.1.0'
    implementation 'com.google.android.material:material:1.1.0'
}
```

How to pass Arraylist to another activity?

3) By using Gson and sharePreference

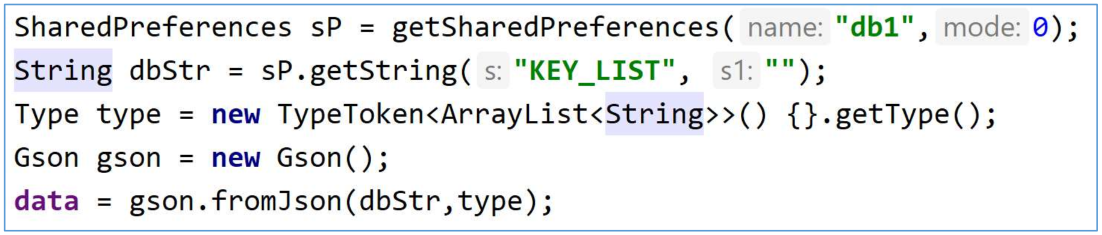
Put it anywhere in your **main activity** that you intent to send to another activity, normally after a button is clicked.



```
String dbStr = gson.toJson(data);

SharedPreferences sP = getSharedPreferences( name: "db1", mode: 0);
SharedPreferences.Editor edit = sP.edit();
edit.putString( s: "KEY_LIST", dbStr);
edit.apply();

Intent i = new Intent( packageContext: this, Main2Activity.class);
startActivity(i);
```

This is your Arraylist

Call this to retrieve your Arraylist in another activity. Then you can get back your array list for other purpose.



```
SharedPreferences sP = getSharedPreferences( name: "db1", mode: 0);
String dbStr = sP.getString( s: "KEY_LIST",  s1: "");
Type type = new TypeToken<ArrayList<String>>() {}.getType();
Gson gson = new Gson();
data = gson.fromJson(dbStr,type);
```

How to pass Arraylist to another activity?

The sample application named “PassArrayList_Malaysia” is available in the supplementary material on Moodle.

Lab time!

Gif retrieved from https://giphy.com/

Lab 6 – Instructions

Task 1:

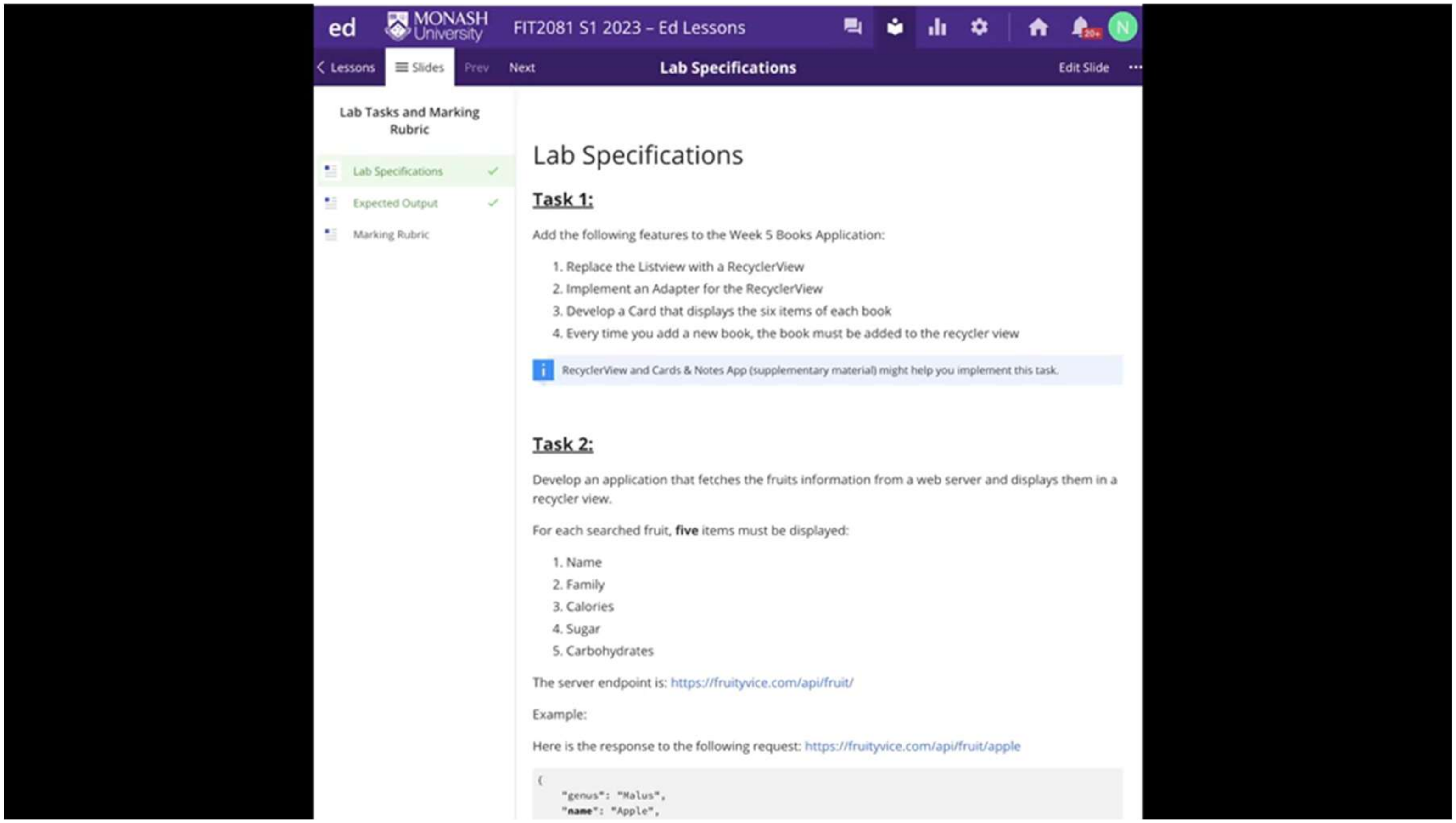
Add the following features to the Week 5 Books Application:

1.Replace the Listview with a RecyclerView
2.Implement an Adapter for the RecyclerView
3.Develop a Card that displays the six items of each book
4.Every time you add a new book, the book must be added to the recycler view

***** RecyclerView and Cards & Notes App (supplementary material) might help you implement this task.**

Lab 6 – Instructions

Task 1:



Lab Specifications

Task 1:

Add the following features to the Week 5 Books Application:

1. Replace the Listview with a RecyclerView
2. Implement an Adapter for the RecyclerView
3. Develop a Card that displays the six items of each book
4. Every time you add a new book, the book must be added to the recycler view

RecyclerView and Cards & Notes App (supplementary material) might help you implement this task.

Task 2:

Develop an application that fetches the fruits information from a web server and displays them in a recycler view.

For each searched fruit, **five** items must be displayed:

1. Name
2. Family
3. Calories
4. Sugar
5. Carbohydrates

The server endpoint is: https://fruityvice.com/api/fruit/

Example:

Here is the response to the following request: https://fruityvice.com/api/fruit/apple

```
{
    "genus": "Malus",
    "name": "Apple",
```

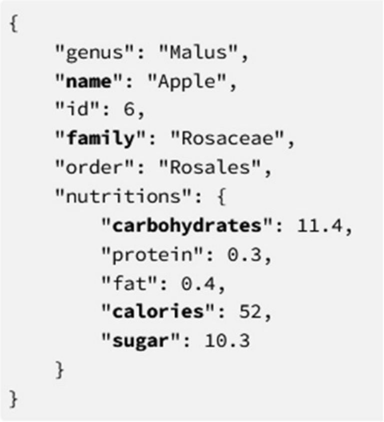
Lab 6 – Instructions

Task 2:
Develop an application that fetches the fruits information from a web server and displays them in a recycler view.

- The app must have one edit text that reads the name of the fruit.
- The app must use RecyclerView to list the results of all searched fruits.
- For each searched fruit, **five** items must be displayed:
  1. Name
  2. Family
  3. Calories
  4. Sugar
  5. Carbohydrates

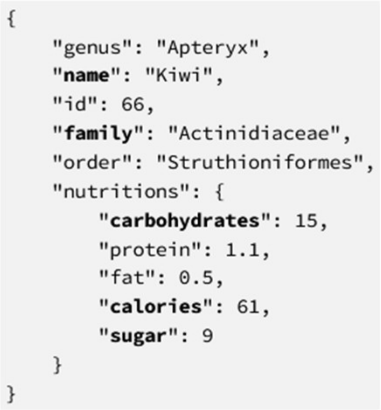
The syntax of the request is: https://fruityvice.com/api/fruit/***FRUIT_NAME***

Example:
Here is the response to the following request:
https://fruityvice.com/api/fruit/apple



```
{
    "genus": "Malus",
    "name": "Apple",
    "id": 6,
    "family": "Rosaceae",
    "order": "Rosales",
    "nutritions": {
        "carbohydrates": 11.4,
        "protein": 0.3,
        "fat": 0.4,
        "calories": 52,
        "sugar": 10.3
    }
}
```

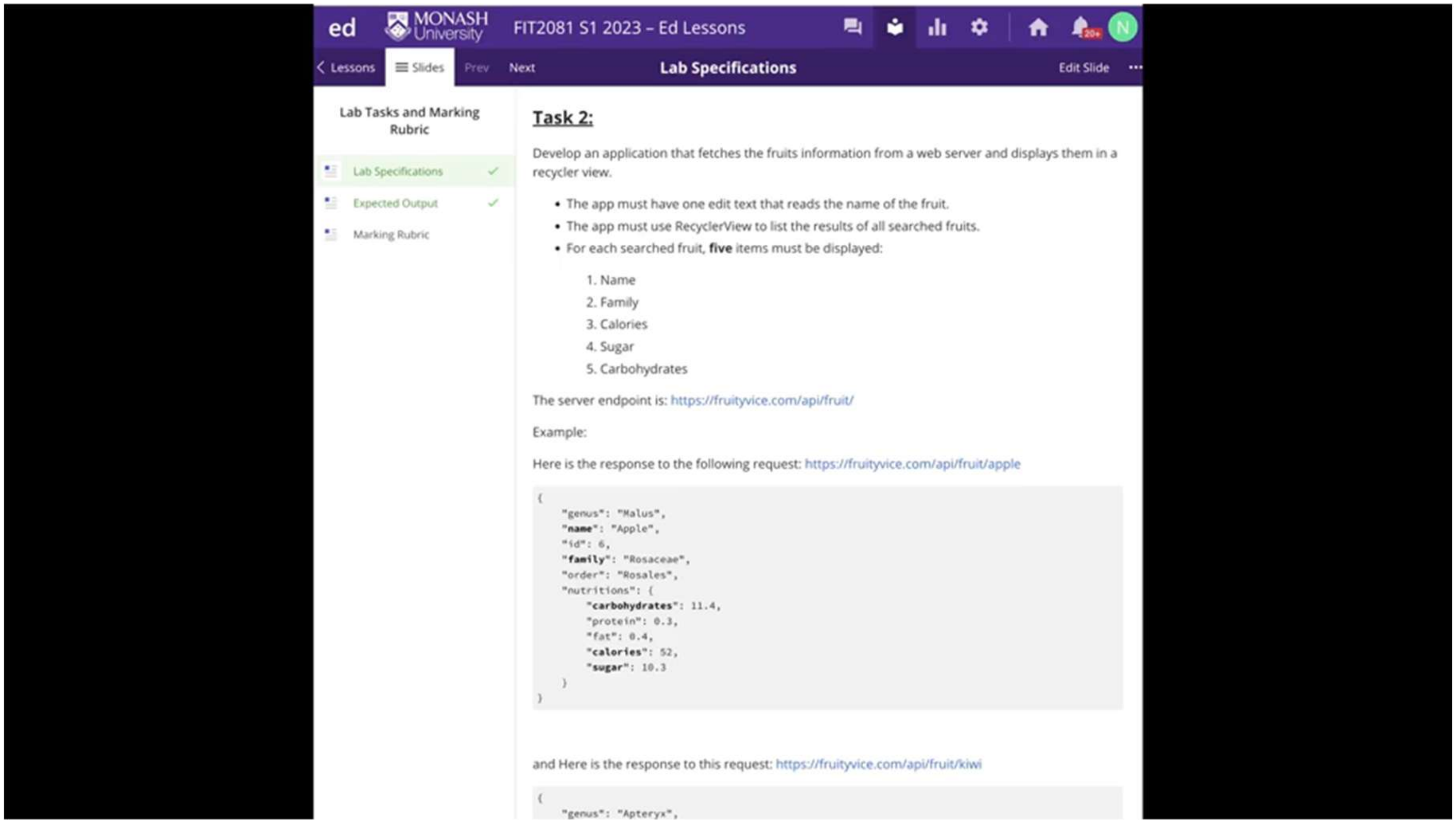
and Here is the response to this request:
https://fruityvice.com/api/fruit/kiwi



```
{
    "genus": "Apteryx",
    "name": "Kiwi",
    "id": 66,
    "family": "Actinidiaceae",
    "order": "Struthioniformes",
    "nutritions": {
        "carbohydrates": 15,
        "protein": 1.1,
        "fat": 0.5,
        "calories": 61,
        "sugar": 9
    }
}
```

Lab 6 – Instructions

Task 2:



Task 2:

Develop an application that fetches the fruits information from a web server and displays them in a recycler view.

- The app must have one edit text that reads the name of the fruit.
- The app must use RecyclerView to list the results of all searched fruits.
- For each searched fruit, **five** items must be displayed:

  1. Name
  2. Family
  3. Calories
  4. Sugar
  5. Carbohydrates

The server endpoint is: https://fruityvice.com/api/fruit/

Example:

Here is the response to the following request: https://fruityvice.com/api/fruit/apple

```
{
    "genus": "Malus",
    "name": "Apple",
    "id": 6,
    "family": "Rosaceae",
    "order": "Rosales",
    "nutritions": {
        "carbohydrates": 11.4,
        "protein": 0.3,
        "fat": 0.4,
        "calories": 52,
        "sugar": 10.3
    }
}
```

and Here is the response to this request: https://fruityvice.com/api/fruit/kiwi

```
{
    "genus": "Apteryx",
```

***Please join your tutorial class if you have any queries regarding the lab tasks.

Thank you!