

## Tutorial 8/Part A of Assignment 2

Due date: 12 October, 13.30 AWST

Marks: 10

This tutorial is part of your assignment 2. Note assignment 2 has four parts and three different due dates. This is the first part and will be marked during tutorial time.

The goal is to make yourself familiar with Android threading and network calls. You will build an app that calls REST services from <https://jsonplaceholder.typicode.com>

You have to use ExecutorService, Callable, and Future.

Functionalities:

1. When the app loads, it performs a Get Request to <https://jsonplaceholder.typicode.com/users>. It loads information about 10 users, such as:

```
[
  {
    "id": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "address": {
      "street": "Kulas Light",
      "suite": "Apt. 556",
      "city": "Gwenborough",
      "zipcode": "92998-3874",
      "geo": {
        "lat": "-37.3159",
        "lng": "81.1496"
      }
    },
    "phone": "1-770-736-8031 x56442",
    "website": "hildegard.org",
    "company": {
      "name": "Romaguera-Crona",
      "catchPhrase": "Multi-layered client-server neural-net",
      "bs": "harness real-time e-markets"
    }
  },
  {
    "id": 2,
    "name": "Ervin Howell",
    "username": "Antonette",
    "email": "Shanna@melissa.tv",
    "address": {
      "street": "Victor Plains",
      "suite": "Suite 879",
      "city": "Wisokyburgh",
      "zipcode": "90566-7771",
      "geo": {
        "lat": "-43.9509",
        "lng": "-34.4618"
      }
    },
    "phone": "010-692-6593 x09125",
    "website": "anastasia.net",
    "company": {
      "name": "Deckow-Crist",
      "catchPhrase": "Proactive didactic contingency",
      "bs": "synergize scalable supply-chains"
    }
  }
]
```

Create a recycler view with the returned usernames. [3 marks]

- The recycler view of the users is clickable; when clicked, we should be able to view the details of the clicked user, i.e., phone, username, address, email, website, and company details. There should be an additional button that will load the posts created by the user. **[2 marks]**
- When the load post button is pressed, we should view all the posts made by the selected user in a recycler view. All posts of the users are available at <https://jsonplaceholder.typicode.com/posts>

```

{
  "userId": 1,
  "id": 9,
  "title": "nesciunt iure omnis dolorem tempora et accusantium",
  "body": "consectetur animi nesciunt iure dolore\nenim quia ad\nveniam autem ut quam aut nobis\net est aut quod aut provident voluptas autem voluptas"
},
{
  "userId": 1,
  "id": 10,
  "title": "optio molestias id quia eum",
  "body": "quo et expedita modi cum officia vel magni\ndoloribus qui repudiandae\nvero nisi sit\nquos veniam quod sed accusamus veritatis error"
},
{
  "userId": 2,
  "id": 11,
  "title": "et ea vero quia laudantium autem",
  "body": "delectus reiciendis molestiae occaecati non minima eveniet qui voluptatibus\naccusamus in eum beatae sit\nvel qui neque voluptates ut commodi qui incidunt\nut animi commodi"
},
{
  "userId": 2,
  "id": 12,
  "title": "in quibusdam tempore odit est dolorem",
  "body": "itaque id aut magnam\npraesentium quia et ea odit et ea voluptas et\nsapiente quia nihil amet occaecati quia id voluptatem\nincidunt ea est distinctio odio"
},
{
  "userId": 2,
  "id": 13,
  "title": "dolorum ut in voluptas mollitia et saepe quo animi",
  "body": "aut dicta possim sint mollitia voluptas commodi quo doloremque\niste corrupti reiciendis voluptatem eius rerum\nsit cumque quod eligendi laborum minima\nperferendis recusandae\narchitecto ipsum ipsam"
},
{

```

However, you must filter out the result based on the selected user id. We should be able to view the details of a post, i.e., title and body. **[5 marks]**

Marking guidelines:

Full marks (100%): all the functionalities are executed without errors with the proper exception handles.

Partial marks (X%+Y%): X% functionalities are executed without errors with the proper exception handles. Y% Coding efforts are there for non-executable parts.

Partial marks (X%): X% functionalities are executed without errors with the proper exception handles. No Coding efforts are there for non-executable parts.

Partial marks (X%-1): X% functionalities are executed without errors; however, no exception handling.

Partial marks (X%-3): X% functionalities are executed without errors; however, no ExecutorService, Callable, and Future.