

## LAB 8 – ANDROID

### GETTING STARTED

The goal of this exercise is to design and implement an Android client for the todo list manager storing tasks with id, description, and urgent.

*Recap:*

1. Fork your own copy of the Git repository associated with this lab (<https://github.com/Aml-2019/android-lab8>) to your personal GitHub space
2. Open Android Studio and select Checkout project from Version Control > Git in the “Welcome to Android Studio” window, to clone your (forked) repository
3. Fill the requested fields (repository URL, location on disk, ...) and press the “Clone” button
4. Commit and push the changes you made back to GitHub, from the VCS menu in Android Studio

### EXERCISE – ANDROID CLIENT

Starting from the base project in the *android-lab8* repository, the goal of this is to design and implement an Android client for the todo list manager storing tasks with id, description, and urgent. The Android application, in particular, will perform HTTP REST requests to the REST Server developed in the Lab 6, in order to show, add, update, and delete tasks<sup>1</sup>.

To perform HTTP requests, you can use the [Spring for Android](#) library, already included in the base project. You can find more information on how to use the library in the [online documentation](#). Please, be aware that every network operation must be execute inside an asynchronous task, by using an [AsyncTask](#) object:

```
new AsyncTask<Void, Void, List<Task>>() {  
  
    @Override  
    protected List<Task> doInBackground(Void... voids) {  
        //here I can perform HTTP requests  
        return RestConsumer.getTasks();  
    }  
  
    @Override  
    protected void onPostExecute(List<Task> tasks){  
        //here I can use the result of the HTTP request, e.g., to  
        //update the user interface  
        if(tasks != null) {  
            //TODO refresh the list of tasks  
        }  
    }  
}.execute();
```

---

<sup>1</sup> To successfully perform HTTP requests, your smartphone must be connected to the same network of the PC hosting the REST server.

The application should be composed of 3 activities:

- The **main activity**, that is opened when the user clicks on the app's icon. It shows the list of all the tasks obtained from the REST server, and a button to add a new task. For displaying a list of elements, you can use the [ListView](#) widget.
- An **add activity**, that is opened when the user clicks on the button for adding a new task. It allows users to define all the fields of a new tasks, and to send the new task to the REST server.
- A **detail activity**, that is opened when the user clicks on a specific task. It shows all the details of the task, and it allows to update and delete the task.