Error Handling and Retry

This lesson will teach you how to handle and recover from errors.

Loading data from the network is not always successful: sometimes there is a weak internet connection which may lead to the timeout error.

It’s always a good idea to handle error scenarios and inform a user about them. If possible, provide ways to recover from the error. In our example, when the loading of the blog article failed, we can show an error message with the “Retry” button to try to reload the data.

To implement error handling, find the `loadData` method of the `BlogDetailsActivity` class.

```
private void loadData() {
    BlogHttpClient.INSTANCE.loadBlogArticles(new BlogArticlesCallback() {
        @Override
        public void onSuccess(List<Blog> blogList) {
            runOnUiThread(() -> showData(blogList.get(0)));
        }

        @Override
        public void onError() {
            // handle error
        }
    });
}
```

All the errors are already delivered to `onError` callback, we just need to handle them. To do so, let’s switch to UI thread via the `runOnUiThread` method and execute the `showErrorSnackbar` method.

```
private void loadData() {
    BlogHttpClient.INSTANCE.loadBlogArticles(new BlogArticlesCallback() {
        @Override
        public void onSuccess(List<Blog> blogList) {
            runOnUiThread(() -> showData(blogList.get(0)));
        }

        @Override
        public void onError() {
            runOnUiThread(() -> showErrorSnackbar());
        }

    });
}
```

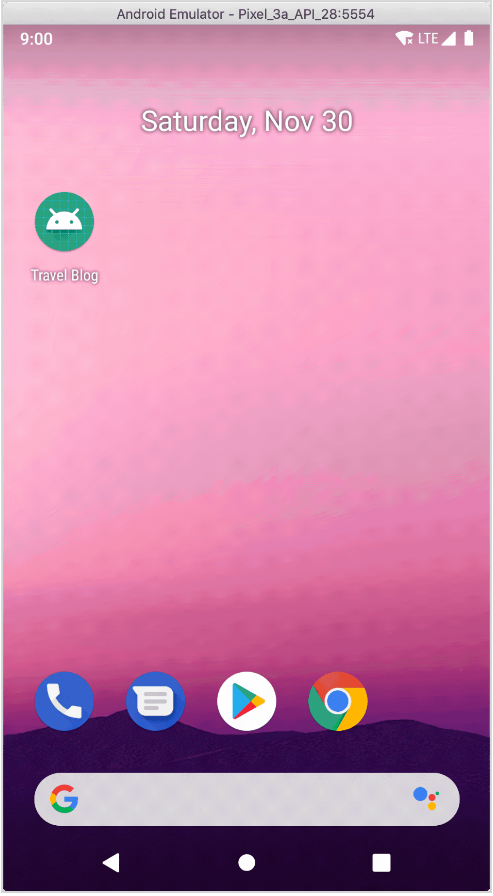
Create the `showErrorSnackbar` method which will be responsible for showing an error message with the “Retry” button.

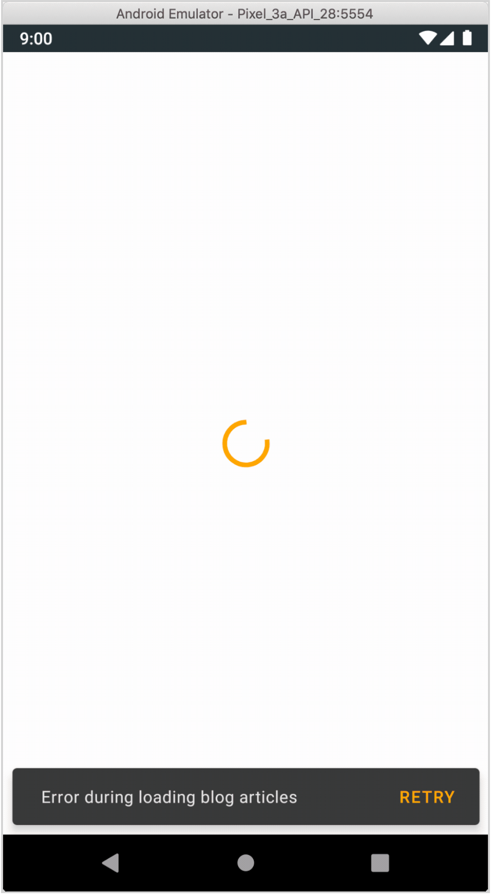
To show the error message, we can use the `Snackbar` component from the material design library:

- to create a `Snackbar` object, use the `Snackbar.make` method which requires three parameters - root view, message, and duration parameters
- to change the color of the “Retry” button we can use the `setActionTextColor` method
- to set the “Retry” button we can use the `setAction` method which requires button name parameter and lambda which will be executed when the button is clicked, in our case we want to execute `loadData` and close the `Snackbar` via `dismiss` method
- to show the `Snackbar`, simply call `show` method

```
private void showErrorSnackbar() {
    View rootView = findViewById(android.R.id.content);
    Snackbar snackbar = Snackbar.make(rootView,
            "Error during loading blog articles", Snackbar.LENGTH_INDEFINITE);
    snackbar.setActionTextColor(getResources().getColor(R.color.orange500));
    snackbar.setAction("Retry", v -> {
        loadData();
        snackbar.dismiss();
    });
    snackbar.show();
}
```

Now, when we launch the application and the error happens, we should see an error message with the “Retry” button.





Error during loading blog articles

Hit the *run* button to try it yourself and don’t forget to turn on “Airplane Mode”.

```
package com.travelblog.http;

public class Author {

    private String name;
    private String avatar;

    public String getName() {
        return name;
    }

    public String getAvatar() {
        return avatar;
    }
}
```

The next lesson will discuss how to render HTML attributes in text.