Project 4 Task 1 – Get Emoji App
Zheng Zeng
ID: zhengzen

Description:

- My application takes a key word from the user and search for the corresponding emoji from Github emoji repository. All the emojis are stored in MongoDB to analyze users' preferences in using the emojis.

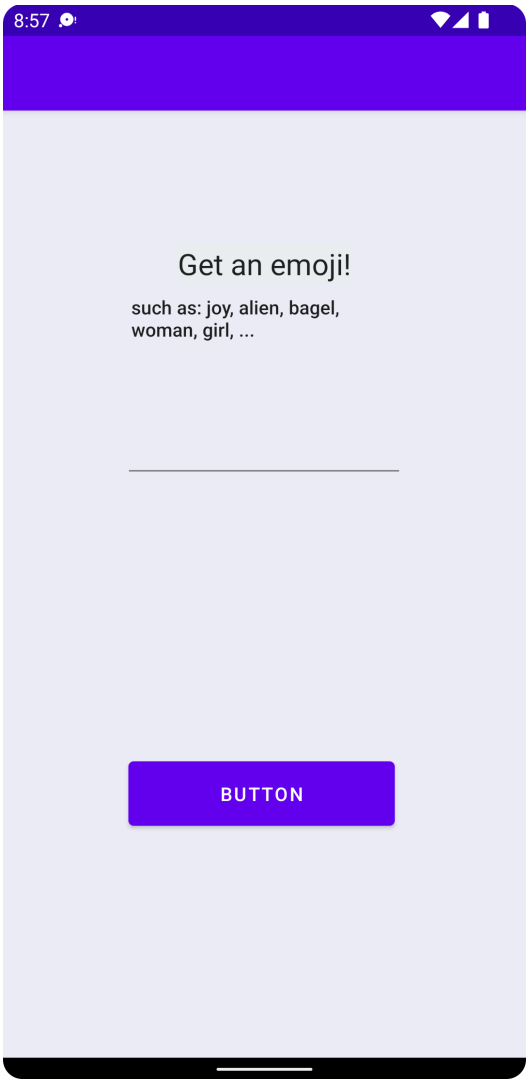
Here is how my application meets the task requirements

1. Implement a native android application

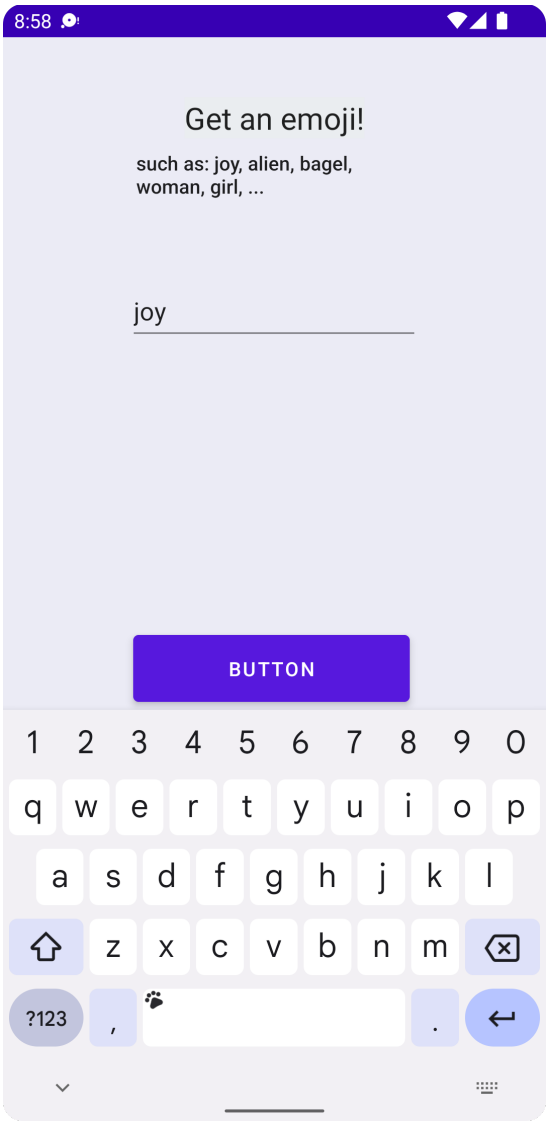
The name of my native android application project in android studio is: getEmoji

1.1 Has at least two different kinds of view:

I provide TextView, ImageView, Button and Plain Text. The details are in the content_main.xml.
Here is the screenshot of the page before fetching any emoji



1.2 Requires input from the user



1.3 makes an HTTP request to my web service

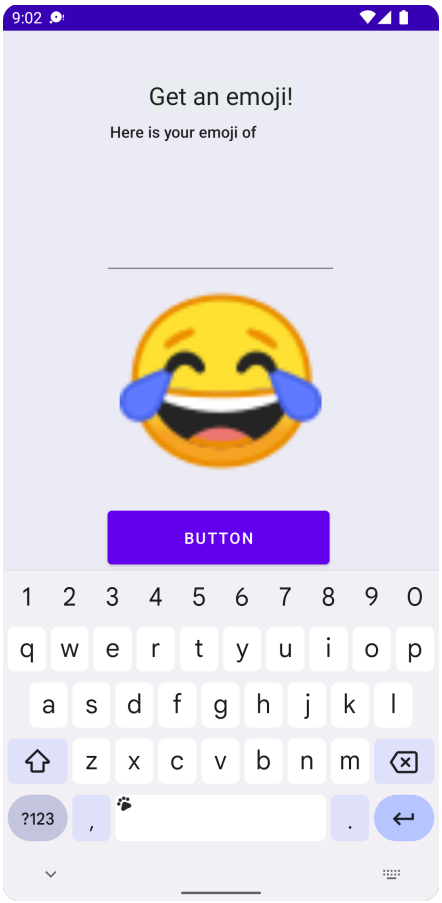
my application does an HTTP GET request in getEmoji.java. the request is https://stormy-everglades-45566.herokuapp.com/?keyword= + keyword

1.4 receive a parse a json file to an Emoji class object

the json file from the web service is

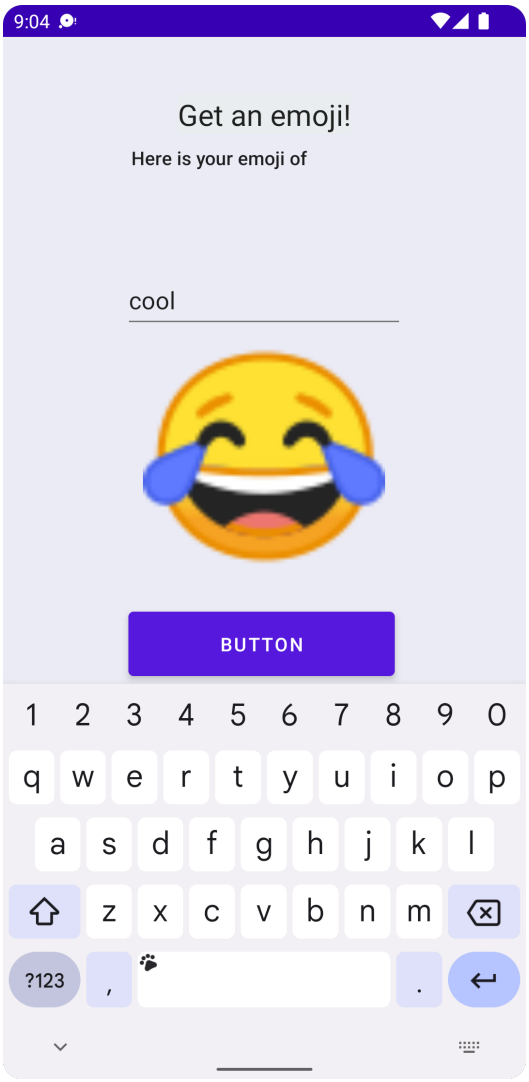
{"image_url":"https://github.githubassets.com/images/icons/emoji/unicode/1f602.png?v8"}

And here is a screenshot of the result



1.5 it is repeatable. After retrieving the first emoji, type in another keyword would retrieve the next one.

Here is a picture of the continuous page



2. implement a web application, deployed to Heroku

the url of my web service deployed to Heroku is : `stormy-everglades-45566`

The project directory name is getEmojiServlet

2.1 using an httpServlet to implement a simple API

in my web app application, the model is getEmoji.java and Emoji.java, and the Controller is getEmojiServlet.java. There is no jsp file created for the application.

2.2 receive an http request from the native Android Application

getEmojiSerlvet.java receives the HTTP GET request with the parameter “keyword” and it passes the key word to the model.

2.3 business logic of my application

The model getEmoji.java first retrieves a list of emojis from the API, and then search for emoji that contains such a key word, and create an Emoji class object to store the image url.

2.4 reply to the android application with a JSON formatted response

The model will use Gson to transform the Emoji class into a JSON-formatted string and return it to the Android application, which will then be parsed by Android’s background thread to convert it into an Emoji class and retrieve its image url.

{"image_url":"https://github.githubassets.com/images/icons/emoji/unicode/1f602.png?v8"}

2.5 Display the image on the application

In the end, the Android background thread will format the image url into a bitmap object, adjust its size to display in and pass it to the UI thread. The UI thread would then position the picture on the page and thus show the emoji.