

Project 4 Task 1 – Interesting Picture App

By Qinzhi Peng

Description:

My mobile app will prompt the user for a string, and use the Pexels API to return a related image tagged with that string and display it in my app.

Here is the Pexels API document: <https://www.pexels.com/api/documentation/>

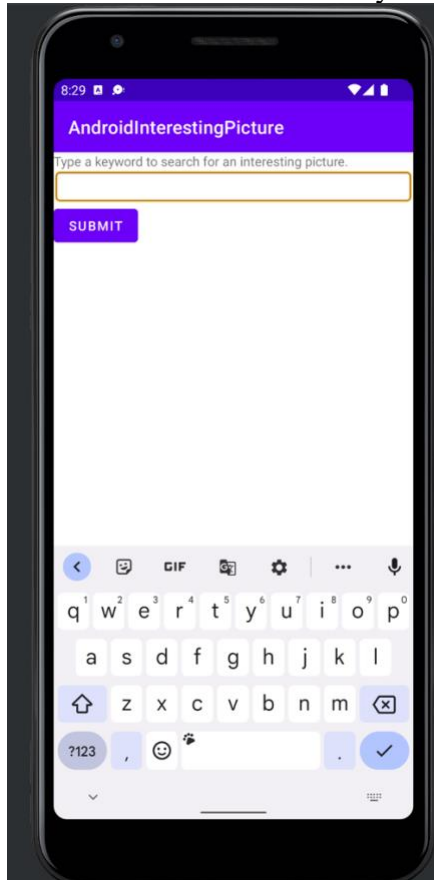
1. Implement a native Android application

The name of my native Android application project in Android Studio is: Project4Android\

1.1. Has at least two different kinds of views in your Layout (TextView, EditText, ImageView, etc.)

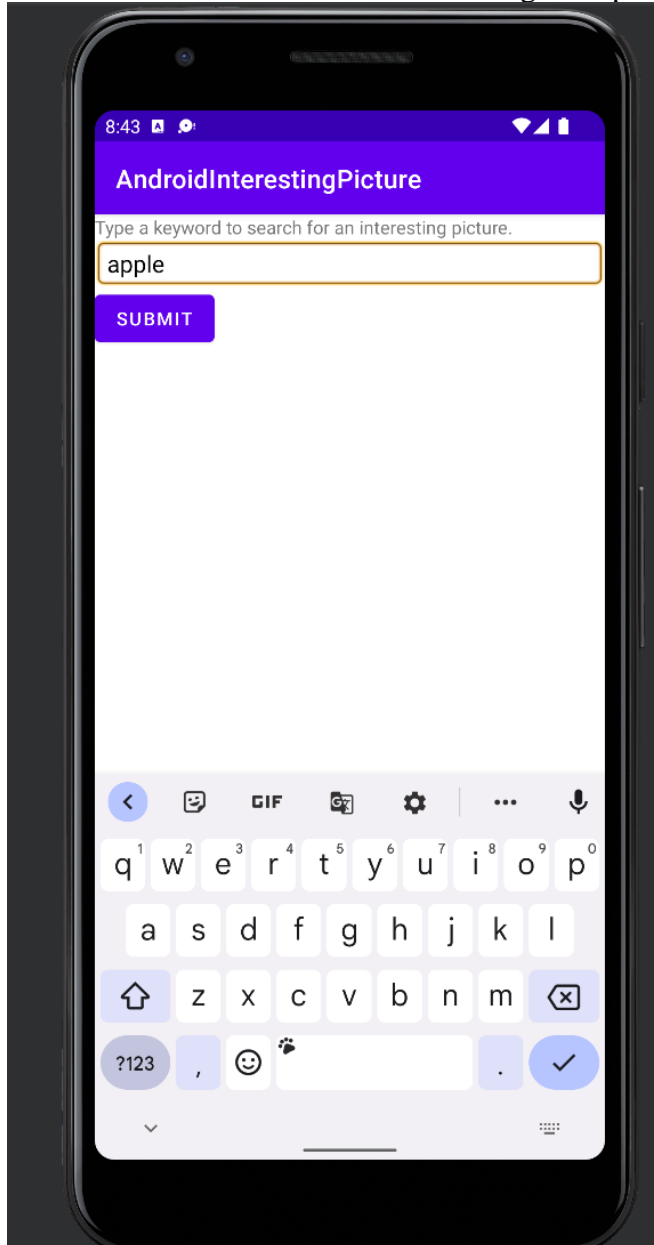
My application uses TextView, EditText, Button, and ImageView. See content_main.xml for details of how they are incorporated into the LinearLayout.

Here is a screenshot of the layout before the picture has been fetched. 1



1.2. Requires input from the user

Here is a screenshot of the user searching for a picture of an apple



1.3. Makes an HTTP request (using an appropriate HTTP method) to your web service

My application does an HTTP GET request in GetPicture.java.

The HTTP request is: "https://guarded-cliffs-

96070.herokuapp.com/getAnInterestingPicture?searchWord=" + searchTerm

where searchTerm is the user's search term.

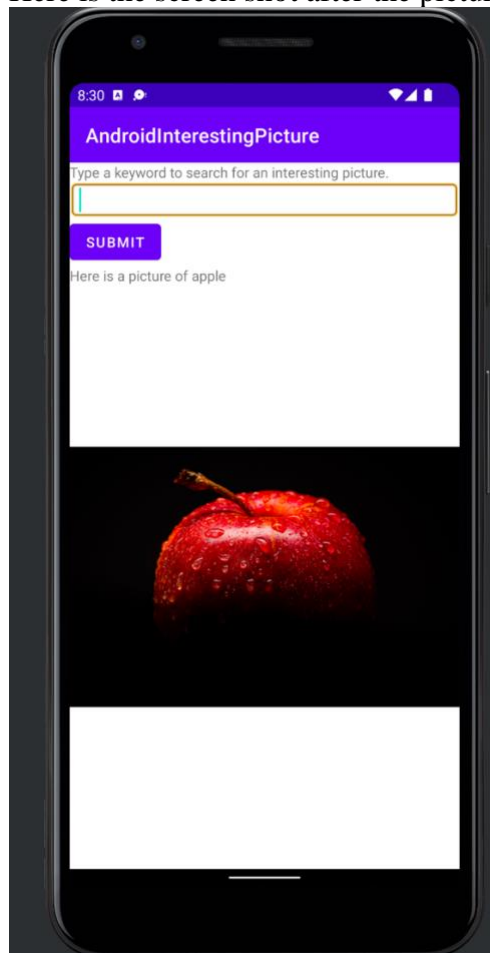
The search method makes this request of my web application, parses the returned XML to find the picture URL, fetches the picture, and returns the image of the picture.

1.4. Receives and parses an XML or JSON formatted reply from the web service
An example of the XML reply is:

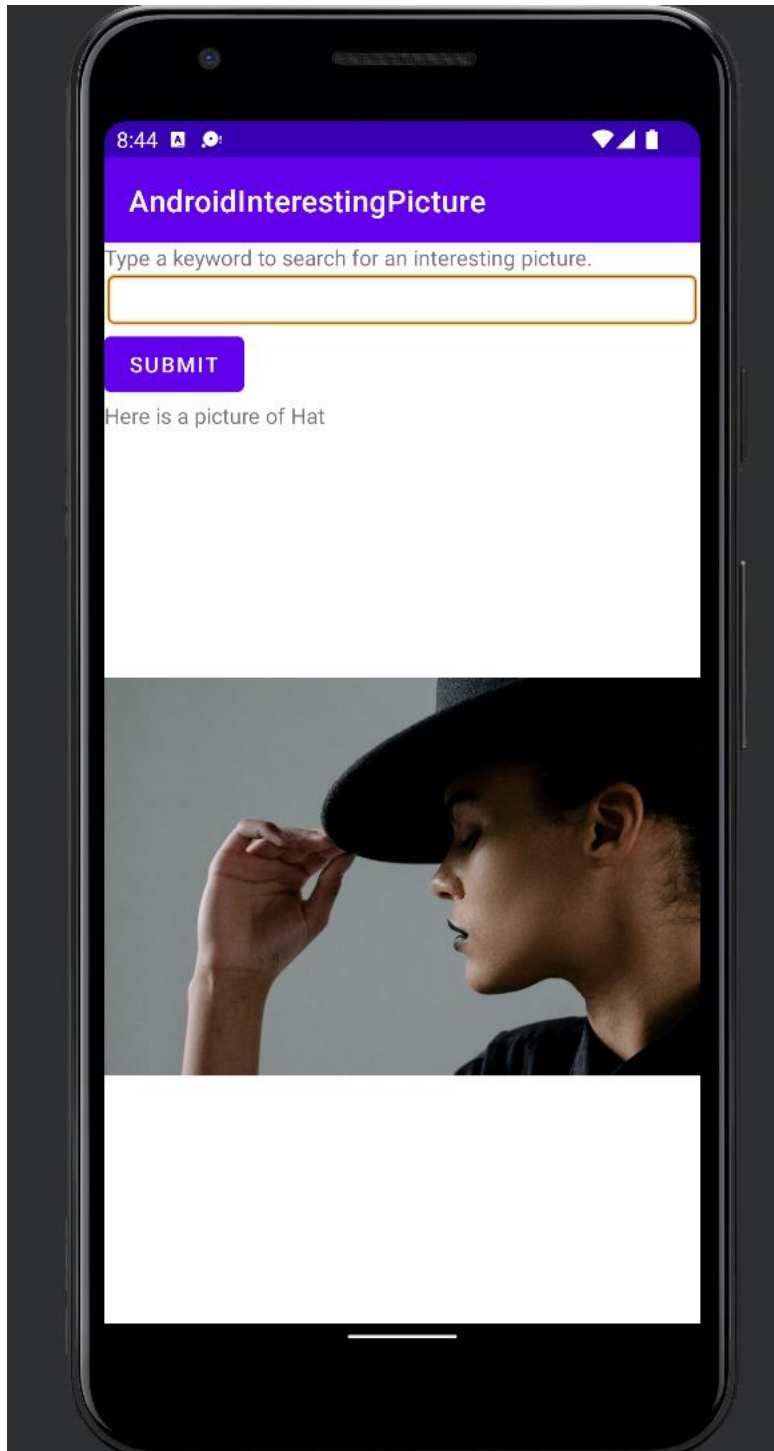
```
<?xml version="1.0" encoding="UTF-8"?>

<html>
<head>
  <title>Interesting Picture</title>
</head>
<body>
  <h1>Here is an interesting picture</h1>
  
  <form action="getAnInterestingPicture" method="GET">
    <label for="letter">Type another word.</label>
    <input type="text" name="searchWord" id="letter" value="" /><br>
    <input type="submit" value="Submit" />
  </form>
</body>
</html>
```

1.5. Displays new information to the user
Here is the screen shot after the picture has been returned.



1.6. Is repeatable (I.e. the user can repeatedly reuse the application without restarting it.)
The user can type in another search term and hit Submit. Here is an example of having typed in "Hat".



2. Implement a web application, deployed to Heroku
The URL of my web service deployed to Heroku is:
guarded-cliffs-96070

2.1. Using an HttpServlet to implement a simple (can be a single path) API

In my web app project:

Model: InterestingPictureModel.java

View: response.jsp , prompt.jsp

Controller: InterestingPictureServlet.java

2.2. Receives an HTTP request from the native Android application

InterestingPictureServlet.java receives the HTTP GET request with the argument "searchWord".
It passes this search searchWord on to the model.

2.3. Executes business logic appropriate to your application

InterestingPictureModelUsingWS.java makes an HTTP request to:

<http://api.flickr.com/services/rest/?method=flickr.photos.search>. It then parses the XM

2.4. Replies to the Android application with an XML or JSON formatted response.

Response.jsp formats the response to the mobile application in a simple XML format of my own design:

```
<%-- @author Qinzhi Peng, qinzhip--%>
<?xml version="1.0" encoding="UTF-8"?>
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<html>
<head>
    <title>Interesting Picture</title>
</head>
<body>
    <% if (request.getAttribute("pictureURL") != null) { %>
    <h1>Here is an interesting picture</h1>
    " alt="random image">
    <% } %>
    <form action="getAnInterestingPicture" method="GET">
        <label for="letter">Type another word.</label>
        <input type="text" name="searchWord" id="letter" value="" /><br>
        <input type="submit" value="Submit" />
    </form>
</body>
</html>
```