

Project 4 Task 2 – CoolNumber App

by Gao Jiang(gaoj)

Description:

My application takes a type and a number from the user, and uses them to fetch and display a special meaning for the number in the given type from <http://numbersapi.com/>. It also supports to fetch a random number meaning for a given type.

Here is how my application meets the task requirements.

1. Implement a native Android application

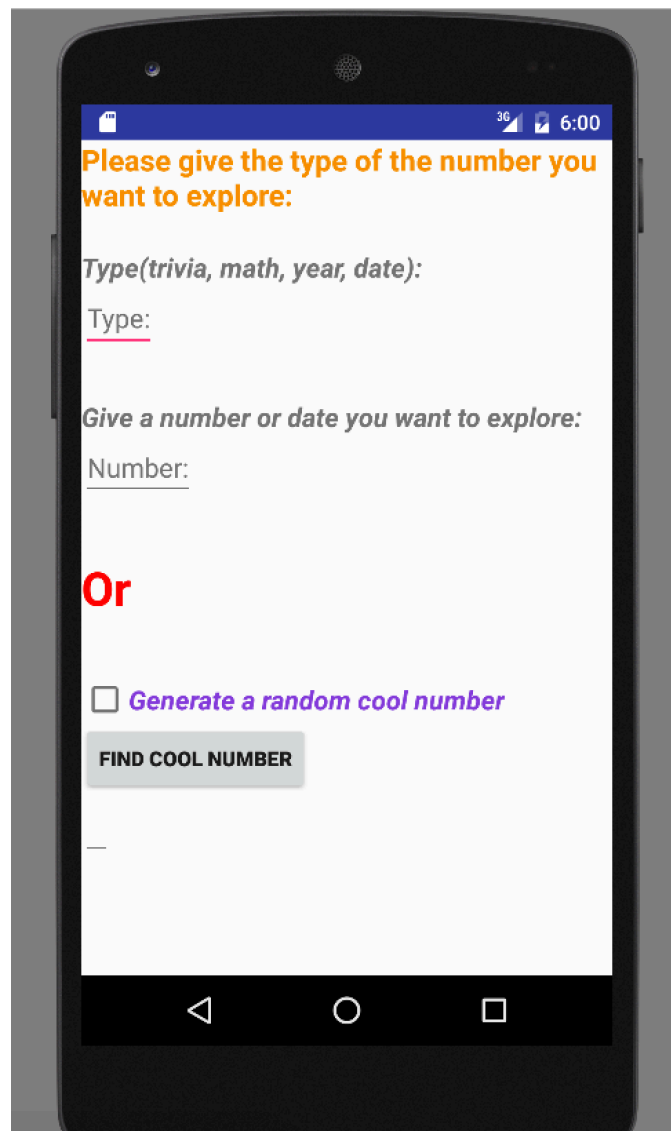
The name of my native Android application project in Android Studio is: CoolNumber

- 1.1. Has at least two different kinds of views in your Layout.

My application uses TextView, EditText and Button.

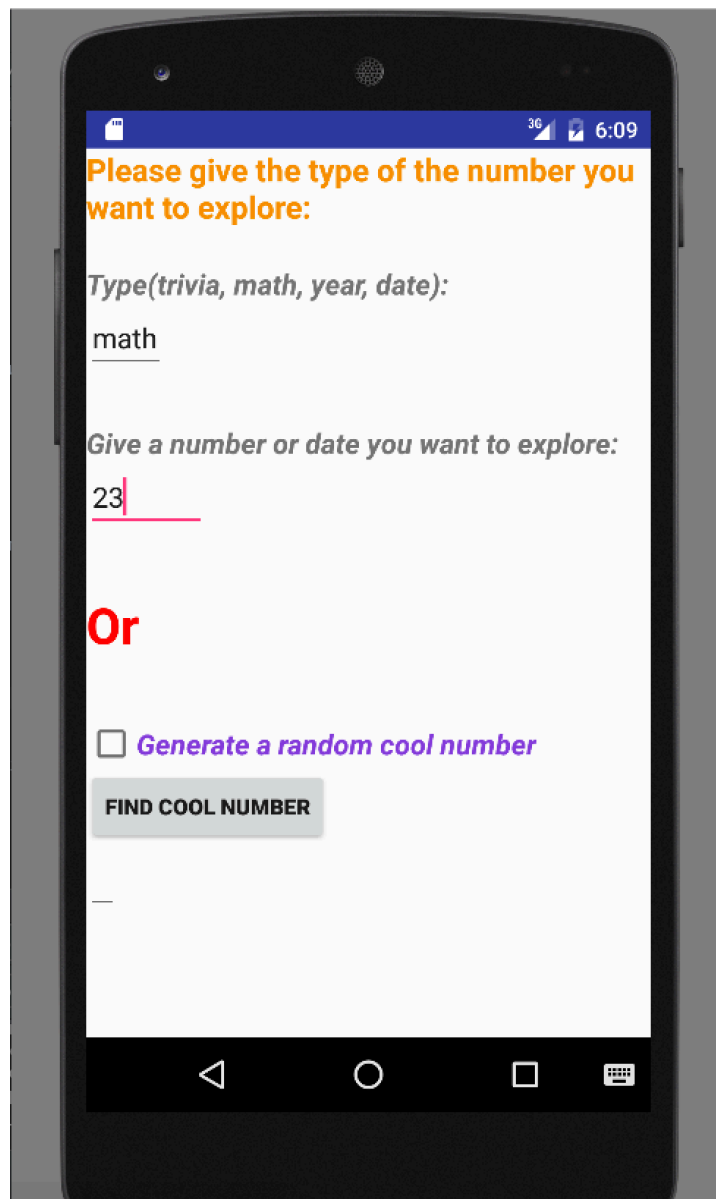
See content_main.xml for details of how they are incorporated into the LinearLayout.

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



- **1.2. Requires input from the user.**

Here is a screenshot of the user searching for number 23 in type math:



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

My application does an HTTP GET request in `GetNumberMeaning.java`. The HTTP request is:

`"https://dry-citadel-3659.herokuapp.com/CoolNumber/" + number + "/" + type"` or
`"https://dry-citadel-3659.herokuapp.com/CoolNumber/random" + "/" + type"`

where number is the user's search number and type is the given search type. If user chooses to generate the meaning randomly, it will use the random parameter. The

search method makes this request of my web application, parses the returned XML to find the cool meaning of the number.

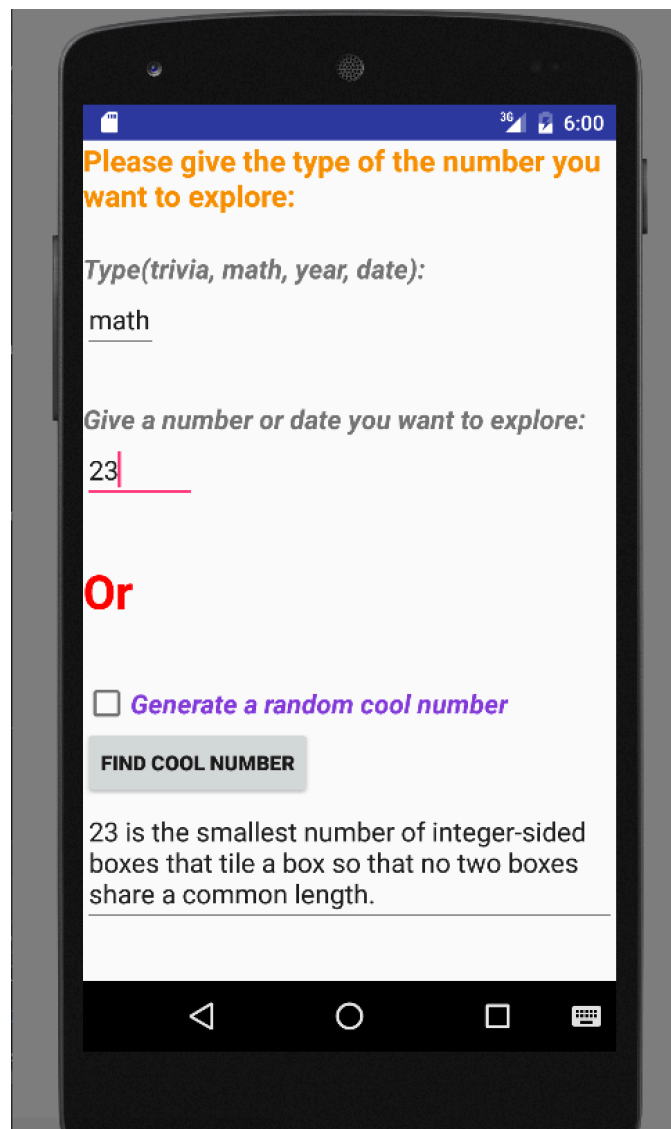
- **1.4. Receives and parses an XML or JSON formatted reply from the web service**
An example of the XML reply is:

```
<CoolNumber>
  <Meaning>
    23 is the smallest number of integer-sided boxes that tile a box so that no two
    boxes share a common length.
  </Meaning>
</CoolNumber>
```

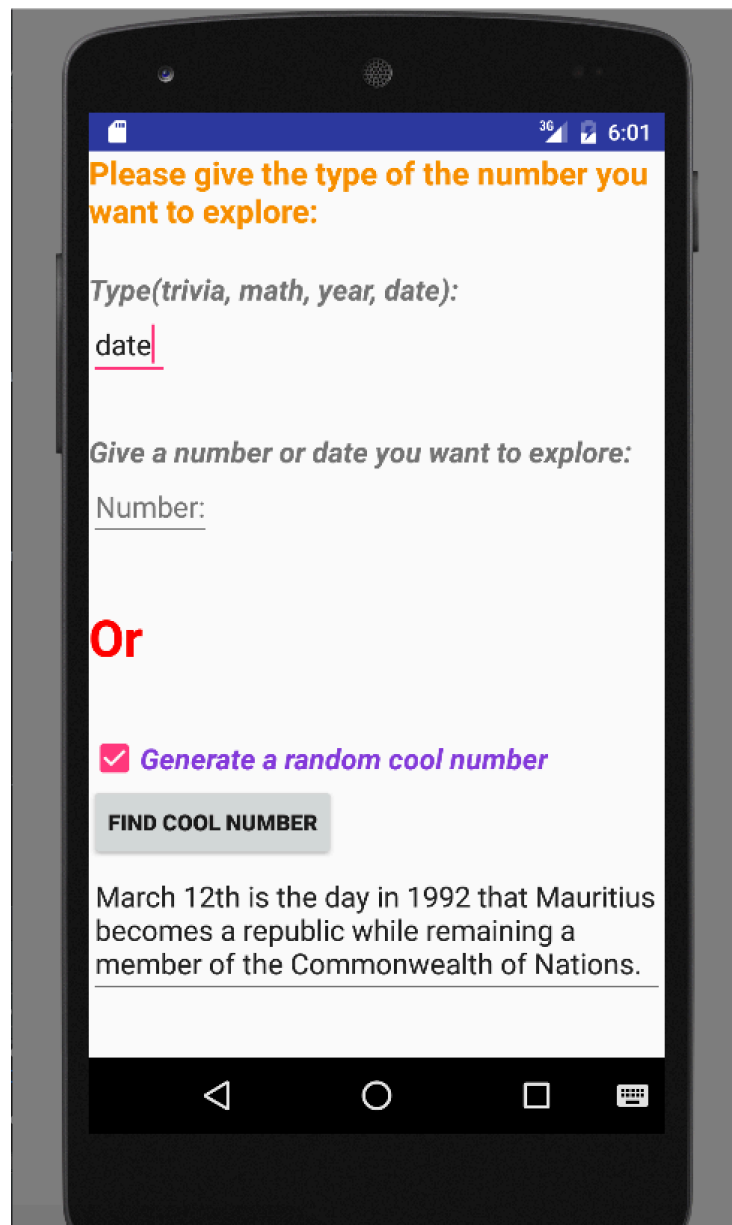
- **1.5 Displays new information to the user**

Here are the screen shots after the meanings has been returned:

Given number and type:



Randomly:

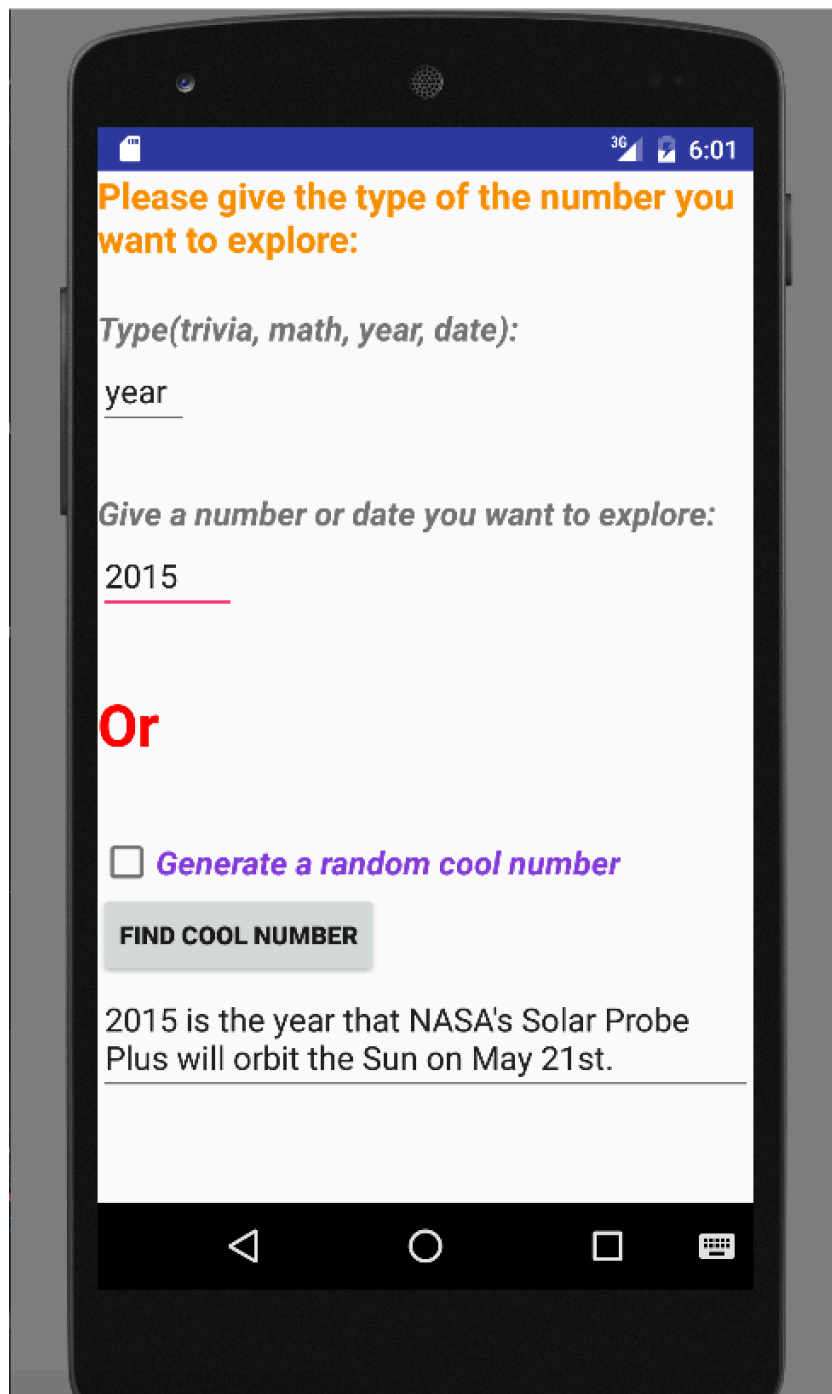


- **1.6. Is repeatable**

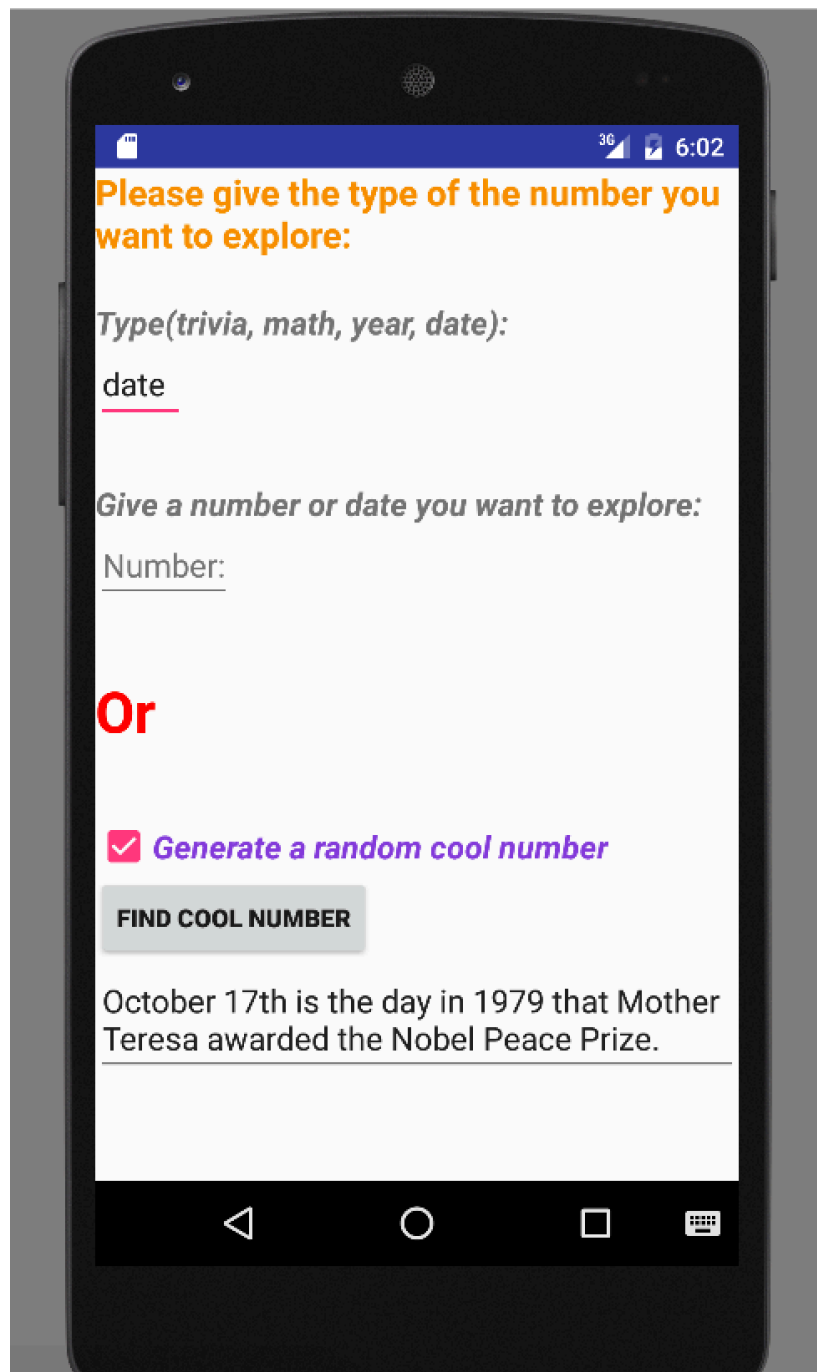
The user can type in another number term and hit Submit.

Here are the screenshots:

Given number and type:



Randomly:



2. Implement a web application, deployed to Heroku

The name of the Heroku project in Android Studio is: CoolNumber
(Heroku deployed app: <https://dry-citadel-3659.herokuapp.com/CoolNumber/>)

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

In my web app project: Servlet: CoolNumber.java

- 2.2. Receives an HTTP request from the native Android application

CoolNumber.java receives the HTTP GET request with the argument number/random and type. It passes parameters and do the search from the API.

- 2.3. Executes business logic appropriate to your application

CoolNumber.java makes an HTTP request to:

<http://numbersapi.com/?number/?type>

It then parses the plain text response and generate a XML response to respond to the Android application.

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

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

```
<CoolNumber>
  <Meaning>
    23 is the smallest number of integer-sided boxes that tile a box so that no two
    boxes share a common length.
  </Meaning>
</CoolNumber>
```