CS 370 Lab 2: Intents and Extras

Join GitHub Team

1. If you haven't already, request permission to join the team for the section you are *enrolled* in.
2. You will not be able to push your lab work if you aren't on a team!

Obtain Code

1. Ensure that the lab workstation is booted into OS X
2. Create a new folder on your desktop called Repositories
3. Open a terminal session (Applications -> Utilities -> Terminal)
    - This is just a terminal window! You **don't** need to log in to blue
4. Ensure there are no other default accounts in the OSX keychain
    a. Keychain management instructions: https://kb.wisc.edu/helpdesk/page.php?id=2197
    b. Search for any *github.com* entries and remove them
5. Using the command line, change directory to the Repositories folder you just created
6. Clone the repository: **git clone https://github.com/SSU-CS370-F18/Android-Lab-2.git**
7. Change directory to the Android-Lab-2 folder you just cloned
8. Branch the repository using a branch name of **lastnamefirstname-370H2**
9. Open Android Studio.
10. Open the Android project that you just cloned.

Basic Intents

1. Expand ***app – res – values***. You should see a file called *dimens.xml*. This resource file holds dimensions in the same way *strings.xml* holds Strings. There is one element defined here:

   ```
   <dimen name="activity_padding">8dp</dimen>
   ```

   This is used to provide some extra spacing along the edges of the window. Look in *activity_main.xml* at the root LinearLayout:

   ```
   android:padding="@dimen/activity_padding"
   ```

   This adds a buffer around the edges of this LinearLayout so that its contents aren't jammed up right against the edge.

2. Using Android Studio, create a new Activity and call it *OtherActivity*

   file → new → Activity → Empty Activity

3. Open *strings.xml* and add these string resources:

   ```
   <string name="welcome_text">Welcome to OtherActivity!"</string>
   <string name="button_text">Go to OtherActivity</string>
   ```

4. Open *activity_other.xml*. Add a TextView to the layout.
   a. Give it an *android:text* of "@string/welcome_text"

   We now have a basic Activity that we can switch to from *MainActivity*.

5. Open *activity_main.xml* and add a Button element
   a. Give it an *android:id* of "@+id/navigate_button"
   b. Give it an *android_text* of "@string/button_text"
6. Open *MainActivity.java*. Add a private Button variable called navigateButton.
7. Inside the *onCreate* method, do the following:
   a. use findViewById to get the *R.id.navigate_button* resource and assign it to *navigateButton*
   b. add an OnClickListener to *navigateButton*:

```
navigateButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(MainActivity.this, OtherActivity.class);
        startActivity(intent);
    }
});
```

The first line in onClick creates an Intent. The second argument is the Activity to start. The second line switches to and starts the Activity defined in the Intent.

8. If your app is running correctly, when you press the button you should see a different screen come up, and it should have text that says **Welcome to OtherActivity!**. If not, revisit the above steps and make sure you've followed the instructions.
9. When your app runs properly, execute the following commands to commit your changes to your branch:
    a. git add .
    b. git commit -m "part 1 complete"

Your code branch is now saved and committed to the *local* git repository. It has not been pushed up to GitHub.

Intents and Passing Information

We can pass information between Activities using the Intent object!

Intents can store **Extra**s, and carry them from one Activity to the next. Each **Extra** you want add requires two things: a **key** and a **value**. The **key** is used to insert and retrieve **value**s from the Intent's collection of **Extra**s.

Open *OtherActivity* and add this line to the class (above *onCreate*)

```
public static final String NAME_KEY = "USER_NAME";
```

This will give us a **key** we can use when putting Extras into an Intent for this Activity. The **value** will be whatever we want to send when we are running the app.

Now let's make some changes to MainActivity so we have some info to pass:

1. Open *activity_main.xml* and add a TextView above the Button
    a. Give it a text of "Enter your name: "
2. Below the TextView and above the Button, add an EditText:
    a. Give it an *android:id* of "@+id/name_edit_text"
    b. Give it an *android:inputType* of "text"

    An EditText is a View that allows the user to type something we can use in the app!
3. Open *MainActivity* and add a private EditText variable called *nameEditText*

4. In *onCreate*, use findViewById to assign the EditText in layout to the *nameEditText* variable
5. Change the navigateButton's listener:

```
@Override
public void onClick(View v) {
    String name = nameEditText.getText().toString();
    Intent intent = new Intent(MainActivity.this, OtherActivity.class);

    intent.putExtra(OtherActivity.NAME_KEY, name);
    startActivity(intent);
}
```

   The first line will retrieve whatever is in the EditText field and save it to *name*.

   The other new line will put the *name* **value** into the Intent, using OtherActivity.NAME_KEY as the **key**.

These changes will let us use the passed information in OtherActivity:

6. Open *activity_other.xml* and add a second TextView below the first.
   a. Give it an *android:id* of "welcome_text"
7. Open *OtherActivity.java*
8. In *onCreate*, add these lines:

```
Intent newIntent = getIntent();
String name = newIntent.getStringExtra(NAME_KEY);
```

   These lines get the Intent that was used to begin this Activity, and retrieve the **value** stored in the **Extra**s using the same **key** we used to insert it in *MainActivity*

9. To protect ourselves if the user clicked the button without entering anything, next add:

```
if(name == null || name.isEmpty()) {
    name = "Not Bob";
}
```

10. Finally, put this at the end to show the information we've passed:

```
welcomeText.setText("Welcome, " + name);
```

11. Add a Button that, when clicked, starts MainActivity. You do not need to pass any Extras.

    The Button's text should say "Return to MainActivity".

12. If your app is running correctly, you should see a text field where you can enter text. Put your name in, and press the button. A new screen should come up that welcomes you by name. If not, revisit the above steps and make sure you've followed the instructions.
13. When your app runs properly, execute the following commands to commit your changes to your branch:
    a. git add .
    b. git commit -m "part 1 complete"
    c. git push origin yourbranchname

Your code branch is now saved and committed to the git repository.

This completes Lab 2.

For bonus points…

Intents and Using Other Apps

You can also use an Intent to control other apps! This section will show you how to open a web page in the user's preferred browser,

1. I will not actually give you bonus points for this part!
2. Especially because I ran out of time to create it!!