The Intent Class

- An Intent is a data structure that specifies
  - An operation to be performed
  - An event that has occurred
- Broadcast by one component
- Received by 0 or more components
- This lectures focuses on using Intents to represent operations, rather than events
### Action

- String representing the operation
- Examples:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Target component</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION_CALL</td>
<td>activity</td>
<td>Initiate a phone call.</td>
</tr>
<tr>
<td>ACTION_EDIT</td>
<td>activity</td>
<td>Display data for the user to edit.</td>
</tr>
<tr>
<td>ACTION_MAIN</td>
<td>activity</td>
<td>Start up as the initial activity of a task, with no data input and no returned output.</td>
</tr>
<tr>
<td>ACTION_SYNC</td>
<td>activity</td>
<td>Synchronize data on a server with data on the mobile device.</td>
</tr>
</tbody>
</table>
Action (cont.)

- Setting the Intent Action

```java
new Intent(Intent.ACTION_VIEW);

Intent newInt = new Intent();
newInt.setAction(Intent.ACTION_VIEW);
```
Data

- Data associated with the Intent
  - Formatted as a Uniform Resource Identifier (URI)
- Examples:
  - Data to view on a map
    - $geo:0,0?q=1600+Pennsylvania+Ave+Washington+DC$
  - Number to dial in the phone dialer
    - $tel:+15555555555$
Setting the Intent data

```java
new Intent(Intent.ACTION_VIEW,
    Uri.parse("tel:+15555555555"));

Intent newInt = new Intent(Intent.ACTION_VIEW);
newInt.setData(Uri.parse("tel:+15555555555"));
```
Additional information about the components that handle the intent

Examples:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATEGORY_BROWNSABLE</td>
<td>The target activity can be safely invoked by the browser to display data referenced by a link — for example, an image or an e-mail message.</td>
</tr>
<tr>
<td>CATEGORY_GADGET</td>
<td>The activity can be embedded inside of another activity that hosts gadgets.</td>
</tr>
<tr>
<td>CATEGORY_HOME</td>
<td>The activity displays the home screen, the first screen the user sees when the device is turned on or when the HOME key is pressed.</td>
</tr>
<tr>
<td>CATEGORY_LAUNCHER</td>
<td>The activity can be the initial activity of a task and is listed in the top-level application launcher.</td>
</tr>
<tr>
<td>CATEGORY_PREFERENCE</td>
<td>The target activity is a preference panel.</td>
</tr>
</tbody>
</table>
Sets the MIME type of the Intent data
  - E.g., “image/*”
- If unspecified, Android will infer the type

Setting the mime type
Intent.setType(String type)
Intent.setDataAndType(Uri data, String type)
- The component to receive this intent
- Setting the component

Intent(Context packageContext, Class<?> cls);

Intent newInt = new Intent();
newInt.setComponent(ComponentName);
newInt.setClass(Context, Class);
### Extras

- **Additional information associated with Intent**
  - Treated as a map (key-value pairs)

- **Setting the Extra attribute**
  - Several forms depending on data types, e.g.,
    - `putExtra(String name, String value);`
    - `putExtra(String name, float[] value);`
- EXTRA_EMAIL: List of email recipients

```java
Intent newInt = new Intent(Intent.ACTION_SEND);
newInt.putExtra(android.content.Intent.EXTRA_EMAIL,
    new String[] {
        aporter@cs.umd.edu, billg@microsoft.com,
        ohhdl@dalailama.com, ladygaga@musician.org
    });
```
Specify how Intent should be handled

Examples:

- FLAG_ACTIVITY_NO_HISTORY
  - Don’t put this Activity in the History stack

- FLAG_DEBUG_LOG_RESOLUTION
  - Causes extra logging information to be printed when this Intent is processed
Intent newInt = new Intent(Intent.ACTION_SEND);
newInt.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);
Using Intents to Activate Activities

- Intents can be used to activate Activities
  - startActivity(Intent intent)
  - startActivityForResult(Intent intent, ...)
- The target Activity can be
  - Named explicitly in the Intent, or
  - Determined implicitly via intent resolution
Explicit Activation

- HelloWorldWithLogin
  - Users must authenticate before they can view the “Hello, Android” screen
- LoginActivity
  - Accepts username & password
  - If password correct, starts HelloAndroid Activity
Activating an Activity (cont.)
public class LoginScreen extends Activity {
    public void onCreate(Bundle savedInstanceState) {
        ...
        loginButton.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                if (checkPassword(uname.getText(), passwd.getText())){
                    Intent helloAndroidIntent =
                        new Intent(LoginScreen.this, HelloAndroid.class);
                    startActivity(helloAndroidIntent);
                }
            }
        });
    }
}
Implicit Activation

- When the Activity to be activated is not named, the system attempts to find Activities that match the Intent
  - Called Intent Resolution
Intent Resolution

- A process for matching Intents with Activities that want to receive them
- Intent Filters describe which Intents an Activity can handle
  - Usually specified in an AndroidManifest.xml file
- Intent Resolution only matches
  - Action
  - Data (both URI and mime data type)
  - Category
<activity ....>
  <intent-filter
    android:icon="drawable resource"
    android:label="string resource"
    android:priority="integer"
    ...
  <action android:name="actionName" />
  ...
</intent-filter>
...
</activity>
Specifying IntentFilters in AndroidManifest.xml

- **android:icon** – Icon representing the activity
- **android:label** - User-readable label for the parent component
- **android:priority** – Priority given to the parent component when handling matching Intents
  - Causes Android to prefer one activity over another
  - Higher values represent higher priorities
Adding data to an IntentFilter

<intent-filter ...>
  ...
  <data android:host="string"
       android:mimeType="string"
       android:path="string"
       android:pathPattern="string"
       android:pathPrefix="string"
       android:port="string"
       android:scheme="string" />
  ...
</intent-filter>
Adding a Category to an IntentFilter

<intent-filter ...>
  ...
  <category android:name="string"/>
  ...
</intent-filter>
Example: Map Application

```
<intent-filter ...>
  <action android:name="android.intent.action.VIEW" />
  <category android:name="android.intent.category.DEFAULT" />
  <category android:name="android.intent.category.BROWSABLE" />
  <data android:scheme="geo"/>
</intent-filter>
```
Note: to receive implicit intents an Activity must specify an IntentFilter with the category
  • "android.intent.category.DEFAULT" category
public class CheckIntents extends Activity {
    ...
    public void onCreate(Bundle savedInstanceState) {
        ...
        checkButton.setOnClickListener(new OnClickListener()
        {
            public void onClick(View v)
            {
                List<String> acts = CheckIntents.getActivitiesForAction(
                    CheckIntents.this,
                    intentText.getText().toString()
                )

                // output results

                }
            });
        }
    }
}
public static List<String> getActivitiesForAction(Context context, String action) {
    final PackageManager packageManager = context.getPackageManager();
    final Intent intent = new Intent(action);
    final List<ResolveInfo> list = packageManager.queryIntentActivities(intent, 0);
    final List<String> acts = new ArrayList<String>();
    for (ResolveInfo ri : list) { acts.add(ri.activityInfo.name); }
    return acts;
}
Show application info
  - % adb shell dumpsys package
Lab Assignment
Source Code Examples

- HelloAndroidWithLogin
- CheckActivityIntent