The Intent Class

The Intent Class

a data structure that represents

- An operation to be performed, or
- An event that has occurred

Intents as Desired Operations

Intents provide a flexible language for specifying operations to be performed

- e.g., Pick a contact, take a photo, dial a phone number

Intents as Desired Operations

Intent is constructed by one component that wants some work done

Received by one activity that can perform that work

Intent Fields

Action
Data
Category
Type
Component
Extras
Flags

Action

String representing desired operation

Examples

ACTION_DIAL – Dial a number

ACTION_EDIT – Display data to edit

ACTION_SYNC – Synchronize device data with server

ACTION_MAIN – Start as initial activity of app

Setting the Intent Action

```
Intent newInt = new!
                Intent(Intent.ACTION_DIAL);
```

Or

```
Intent newInt = new Intent();
newInt.setAction(Intent.ACTION_DIAL);
```

Data

- Data associated with the Intent
  - Formatted as a Uniform Resource Identifier (URI)

Examples

Data to view on a map

```
Uri.parse("geo:0,0?                    !
        q=1600+Pennsylvania+!
        Ave+Washington+DC")
```

Number to dial in the phone dialer

```
Uri.parse("tel:+1555555555")
```

Setting Intent data

```
Intent newInt = new Intent ( !
                Intent.ACTION_DIAL,!
                Uri.parse("tel:+15555555555"));
```

Or

```
Intent newInt = !
                new Intent(Intent.ACTION_DIAL);
newInt.setData(!
                Uri.parse("tel:+15555555555"));
```

Category

Additional information about the components that can handle the intent

Examples

Category_browsable – can be invoked by a browser to display data ref's by a URI

Category_launcher – can be the initial activity of a task & is listed in top-level app launcher

Type

Specifies the MIME type of the Intent data

Examples

image/*, image/png, image/jpeg

text/html, text/plain

If unspecified, Android will infer the type

Setting the type

Intent.setType(String type)

Or

Intent.setDataAndType(Uri data,!
String type)

Component

The component that should receive this intent

Use this when there’s exactly one component that should receive the intent

Setting the component

```
Intent newInt = Intent(!
        Context packageContext, Class<?> cls);
```

Setting the component

Or

Intent newInt = new Intent ();

and one of:

setComponent(), setClass(), or setClassName()

Extras

Add’l information associated with Intent

Treated as a map (key-value pairs)

Examples

Intent.EXTRA_EMAIL: email recipients

```
Intent newInt = new Intent(Intent.ACTION_SEND);
newInt.putExtra(android.content.Intent.EXTRA_EMAIL,
  new String[]{!
          “aporter@cs.umd.edu”, “ceo@microsoft.com”,!
            “potus@whitehouse.gov”,“mozart@musician.org”
           }
);
```

Setting the Extra attribute

Several forms depending on data type

putExtra(String name, String value);

putExtra(String name, float[] value);

...

Flags

Specify how Intent should be handled

Examples

FLAG_ACTIVITY_NO_HISTORY

Don’t put this Activity in the History stack

FLAG_DEBUG_LOG_RESOLUTION

Print extra logging information when this Intent is processed

Setting Flags

```
Intent newInt = !
                new Intent(Intent.ACTION_SEND);
newInt.setFlags(!
        Intent.FLAG_ACTIVITY_NO_HISTORY);
```

Starting Activities with Intents

startActivity(Intent intent,...)

startActivityForResult(Intent intent, ...)

The target Activity

Can be named explicitly by setting the intent’s component

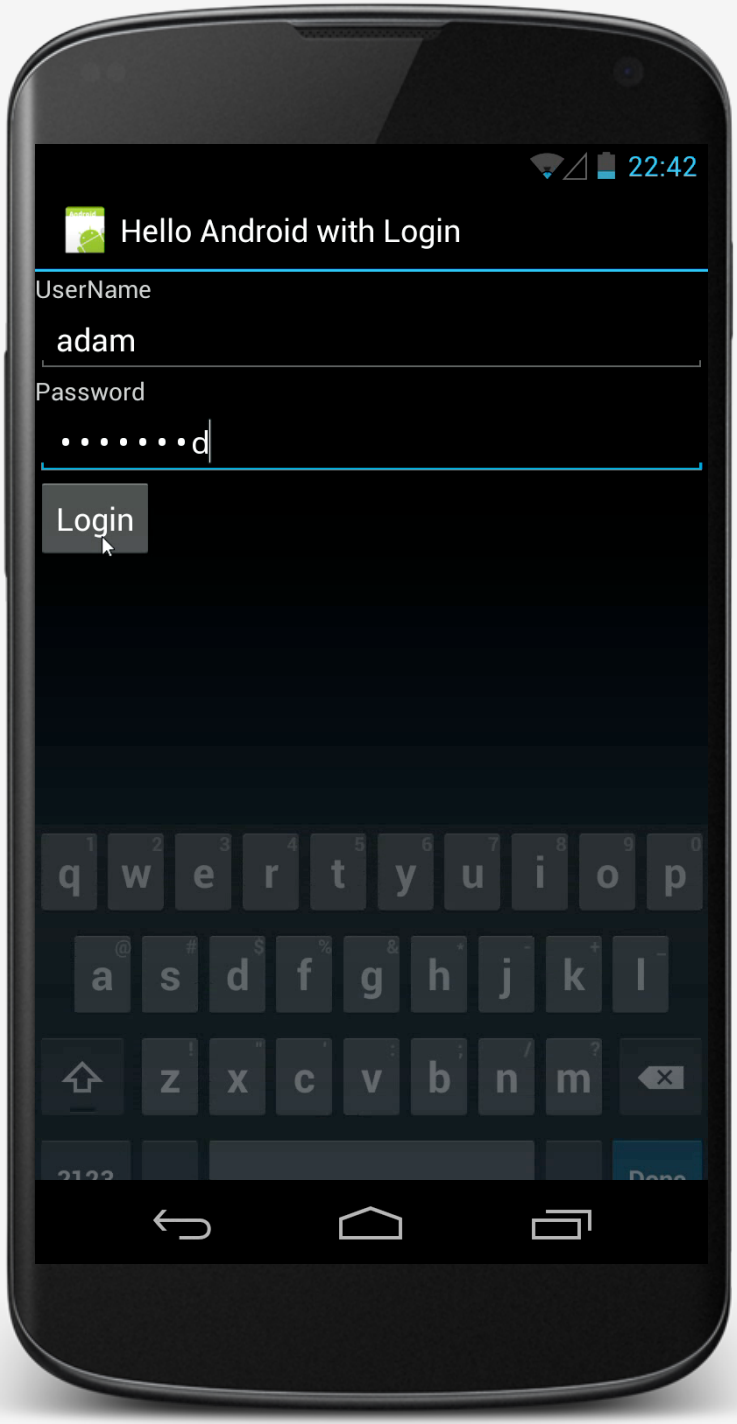
Can be determined implicitly

Explicit Activation

HelloWorldWithLogin

two Activities

- LoginActivity checks username & password and then starts HelloAndroidActivity
- HelloAndroidActivity shows “hello Android” message


22:42
Hello Android with Login
UserName
adam
Password
•••••••d
Login

HelloWorldWithLogin

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.loginscreen);

    final EditText uname = (EditText) findViewById(R.id.username_edittext);
    final EditText passwd = (EditText) findViewById(R.id.password_edittext);

    final Button loginButton = (Button) findViewById(R.id.login_button);
    loginButton.setOnClickListener(new OnClickListener() {

        public void onClick(View v) {

            if (checkPassword(uname.getText(), passwd.getText())) {

                // Create an explicit Intent for starting the HelloAndroid Activity
                Intent helloAndroidIntent = new Intent(LoginScreen.this,
                        HelloAndroid.class);

                // Use the Intent to start the HelloAndroid Activity
                startActivity(helloAndroidIntent);

            } else {
                uname.setText("");
                passwd.setText("");
            }
        }
    });
}
```

Implicit Activation

When the Activity to be activated is not explicitly named, Android tries to find Activities that match the Intent

This process is called intent resolution

Intent Resolution Process

An Intent describing a desired operation

IntentFilters which describe which operations an Activity can handle

- Specified either in AndroidManifest.xml or programmatically

Intent Resolution Data

Action

Data (both URI & TYPE)

Category

Specifying IntentFilters

```
<activity ...>
  <intent-filter ...>!
      ...
      <action android:name=“actionName"/>
      ...
  </intent-filter>
...
</activity>
```

Handling Intent.ACTION_DIAL

```
<activity ...>
  <intent-filter ...>
      ...
      <action android:name=!
            ”android.intent.action.DIAL" />
      ...
  </intent-filter>
...
</activity>
```

Adding data to IntentFilter

```
<intent-filter ...> !
    ...
    <data
      android:mimeType="string"
      android:scheme="string" !
      android:host="string"
      android:port="string"
      android:path="string"
      android:pathPattern="string"
      android:pathPrefix="string"
/>!
    ...
</intent-filter>
```

See: http://developer.android.com/guide/!components/intents-filters.html

Handling geo: scheme Intents

```
<intent-filter ...> !
    ...
    <data android:scheme=”geo" />!
    ...
</intent-filter>
```

Adding A Category to IntentFilter

```
<intent-filter ...>!
    ...
    <category android:name="string" />!
    ...
</intent-filter>
```

Example: Maps Application

Example: Maps 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>!
```

Receiving Implicit Intents

Note: to receive implicit intents an Activity should specify an IntentFilter with the category

"android.intent.category.DEFAULT”

Priority

android:priority – Priority given to the parent component when handling matching Intents

Causes Android to prefer one activity over another

Value should be greater than -1000 & less than 1000

Higher values represent higher priorities