The Internet

CS 2046

Mobile Application Development

Fall 2010

Announcements

- HW2 due Monday, 11/8, at 11:59pm
- If you want to know what it means to “root” your phone, or what this does, see Newsgroup
- HW1 grades on CMS
  - Solutions coming soon
  - Make sure you read feedback!

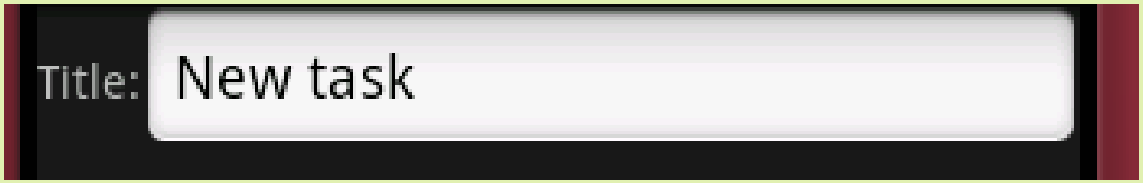
Common HW1 Mistakes

- Compiler warnings
  - Unused variables
    - Probably means you've missed something
- Layout bugs
  - Overuse of android:layout_weight
    - Don't forget about fill_parent and wrap_content!
  - Absolute values
    - Even if they're in dips, unneeded for this simple layout

TaskEdit Layout (One Approach)

- Everything goes in a vertical LinearLayout

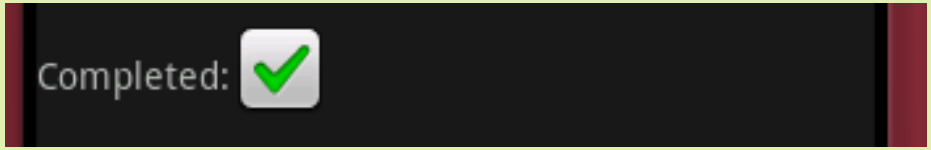
Row 1:




```
<LinearLayout android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:orientation="horizontal">
  <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/edit_title"/>
  <EditText android:id="@+id/edit_title"
            android:layout_width="0px"
            android:layout_weight="1"
            android:layout_height="wrap_content" />
</LinearLayout>
```

TaskEdit Layout

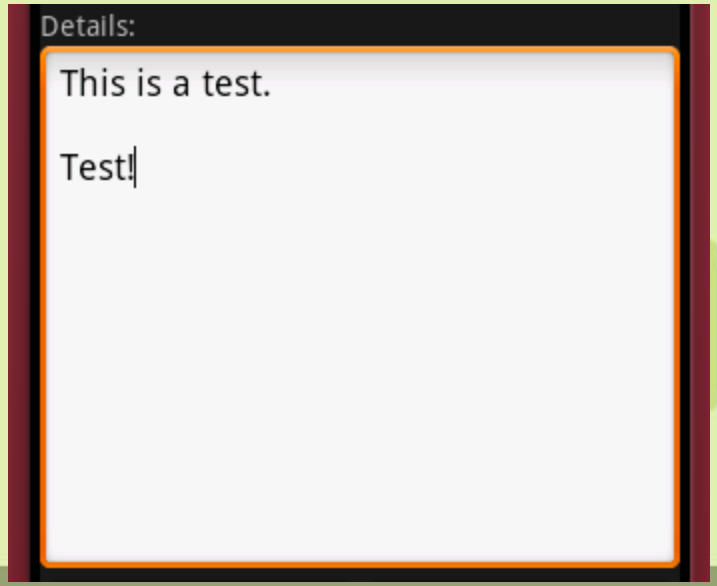
- Row 2:




```
<LinearLayout android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:orientation="horizontal">
  <TextView android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/edit_completed"/>
  <CheckBox android:id="@+id/edit_completed"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
</LinearLayout>
```

TaskEdit Layout

```
<TextView android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:text="@string/edit_details"/>
<EditText android:id="@+id/edit_details"
          android:layout_width="fill_parent"
          android:layout_height="0px"
          android:layout_weight="1"
          android:scrollbars="vertical"
          android:gravity="top"/>
```



TaskEdit Layout

```
<LinearLayout android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:orientation="horizontal">
  <Button android:id="@+id/save_task"
          android:layout_width="0px"
          android:layout_height="wrap_content"
          android:layout_weight="1"
          android:text="@string/save_task" />
  <Button android:id="@+id/delete_task"
          android:layout_width="0px"
          android:layout_height="wrap_content"
          android:layout_weight="1"
          android:text="@string/delete_task" />
</LinearLayout>
```

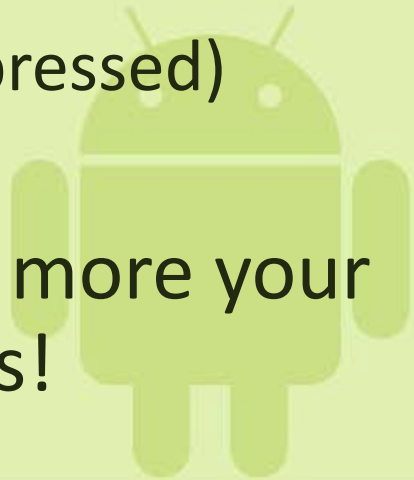
Recap

- BroadcastReceivers and Services

- Today – something you might need to do in the background.
  - Accessing the Internet

Mobile Internet

- Connections vary in:
  - Speed
  - Reliability
  - Availability
- Minimize the amount of data transmitted online.
  - Store HTML pages as a local resource
  - Keep downloaded data small (and/or compressed)
- Keep in mind – for some data plans, the more your app downloads, the more your user pays!



HTML Pages

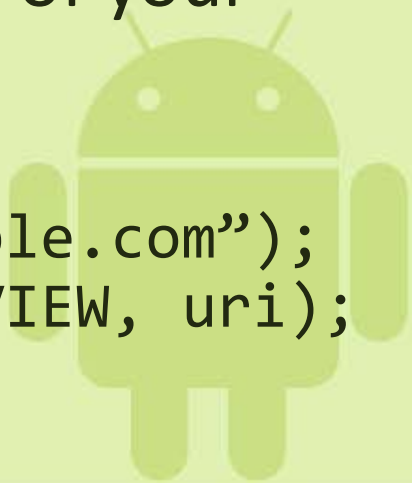
- Useful for:
  - Content which can be updated without having to update the program.
  - Anything which might be laid out more easily with HTML
    - Anything text-heavy (help files)
  - Content which can be used across different mobile platforms.
    - Useful if porting to iOS or elsewhere

WebView

- View which displays web pages
  - Uses WebKit engine
    - Safari, Chrome
  - Loads pages in background thread automatically
  - Ranges from extremely dumb (displays a static webpage) to very complex (a full browser)

- Is the page you're trying to view really part of your application?
  - Consider launching Browser instead with:

```
Uri uri = Uri.parse("http://www.example.com");
Intent i = new Intent(Intent.ACTION_VIEW, uri);
startActivity(i);
```



Internet Permission

- Needed in Manifest:

```
<uses-permission android:name="android.permission.INTERNET" />
```

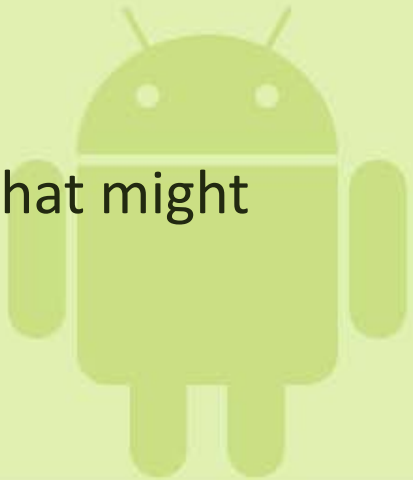
- If you forget this, you may not get a security exception.
  - On WebViews – all pages say “Web page not available”, no exception is thrown.

WebView

- Like any other widget
  - Can use <WebView> tag in XML layout
- Simplest settings:
  - No widgets for browser activity (back, forward, etc.)
  - Javascript disabled
  - Clicking links will launch the browser

Basic WebView

- Places we can affect behavior:
  - Create and set a WebChromeClient subclass
    - Set with setWebChromeClient()
    - Methods are called when something happens that might impact the browser UI
      - Progress updates, JavaScript alerts, page titles
  - Create and set a WebViewClient subclass
    - Set with setWebViewClient()
    - Methods are called when something happens that might affect the rendering of page content
      - Errors, form submissions, URL loading

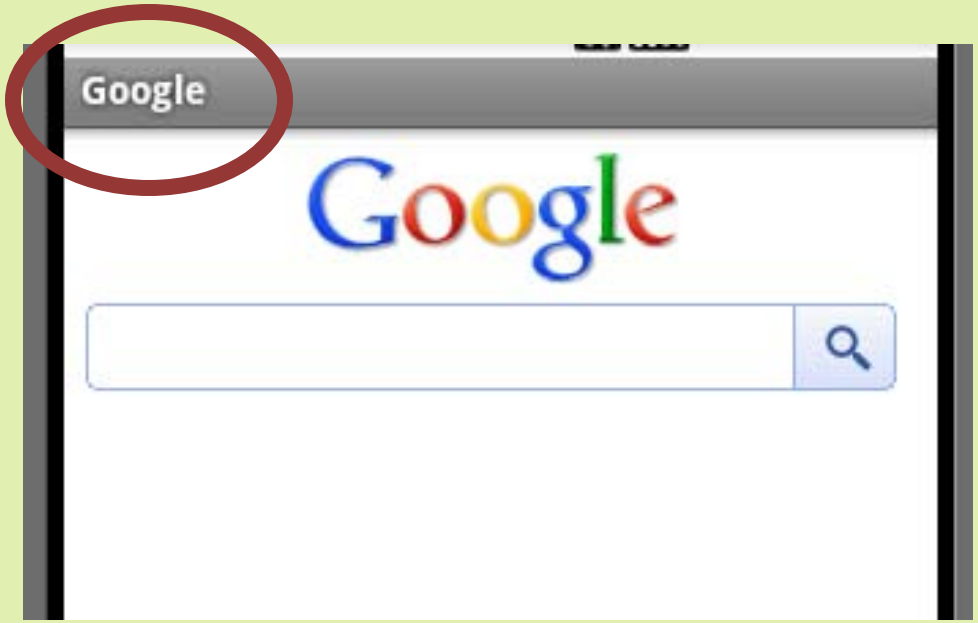


Basic WebView

- Places we can affect behavior:
  - Modify the WebSettings
    - Access with getSettings()
    - Enable JavaScript, zoom controls, fonts, other basic WebView properties
  - Add JavaScript-Java interfaces
    - Use addJavascriptInterface()
    - Allows JavaScript from the displayed webpage to modify the Views in your Activity.
- See class documentation for all that can be done.

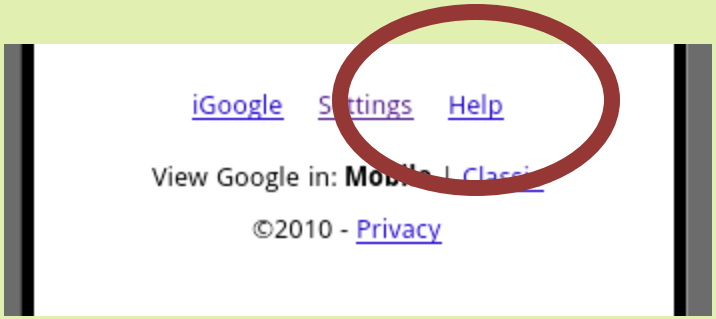
Using WebChromeClient

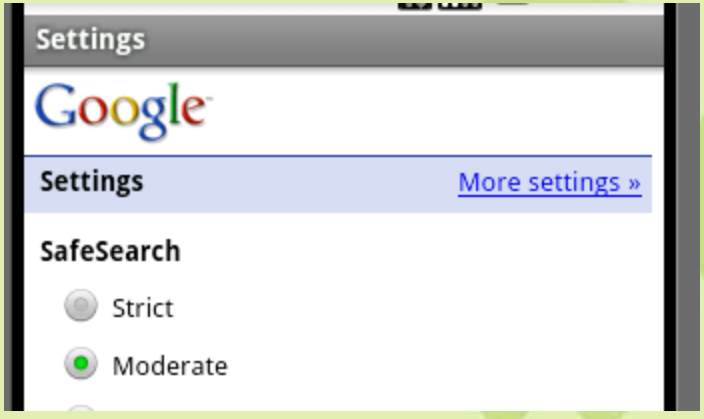
```
wv.setWebChromeClient(new WebChromeClient() {
    @Override
        public void onReceivedTitle(WebView view,
                String title) {
            getWindow().setTitle(title);
        }
    });
```



Using WebViewClient

```
wv.setWebViewClient(new WebViewClient() {
    @Override
    public boolean shouldOverrideUrlLoading(WebView view,
            String url) {
        wv.loadUrl(url);
        return false;
    }
});
```





Modifying WebSettings

```
WebSettings settings = wv.getSettings();
settings.setDefaultFontSize(20);
settings.setBuiltInZoomControls(true);
settings.setJavaScriptEnabled(true);
```

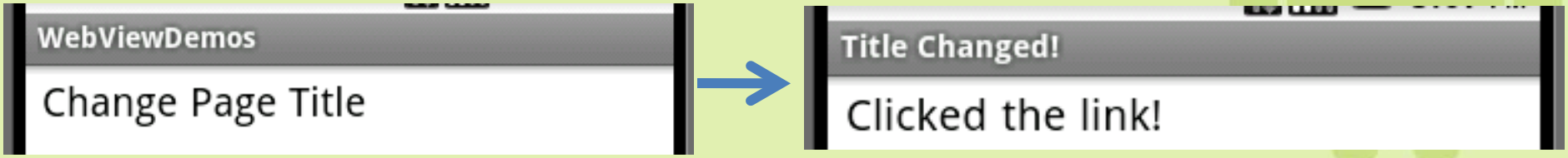
1 Google.com on your Android device
2 Mobile Search and iGoogle
3 Local Search
4 Product Search

1 Google.com on your Android device
2 Mobile Search and iGoogle
3 Local Search
4 Product Search

JavaScript – Java Interface

```
wv.addJavascriptInterface(new JSInteraction(), "webviewjs");
wv.loadData("
  <a onClick=\"window.webviewjs.updateTitle()\">
    Change Page Title
  </a>", "text/html", "utf-8");

public class JSInteraction {
    public void updateTitle() {
        // Handler needed to run this on UI thread.
        mHandler.post(new Runnable() {
            public void run() {
                getWindow().setTitle("Title Changed!");
                wv.loadData("Clicked the link!", "text/html", "utf-8");
            }
        });
    }
}
```



Back Button

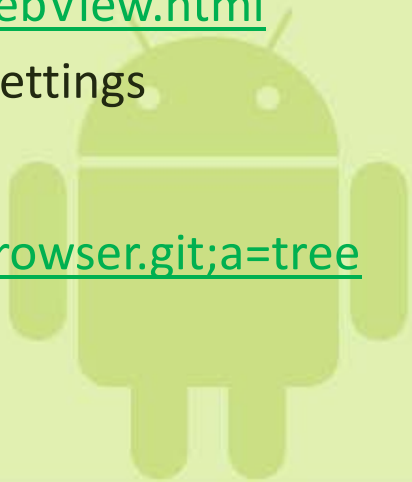
- Just like any other Android view
  - In Activity containing it, can override onKeyDown:

```
@Override
public boolean onKeyDown(int keyCode,
        KeyEvent event) {
    if ((keyCode == KeyEvent.KEYCODE_BACK)
            && wv.canGoBack()) {
        wv.goBack();
        return true;
    }
    return super.onKeyDown(keyCode, event);
}
```

WebView Summary

- What we've seen only scratches the surface…
  - Managing cookies
  - Progress notifications

- Where to look if you want to do more:
  - WebView: http://developer.android.com/reference/android/webkit/WebView.html
    - Don't forget – WebViewClient, WebChromeClient, WebSettings
  - Android Browser source code: http://android.git.kernel.org/?p=platform/packages/apps/Browser.git;a=tree



Downloading over HTTP

- Uses standard Java library
  - For more complex needs, see HttpClient

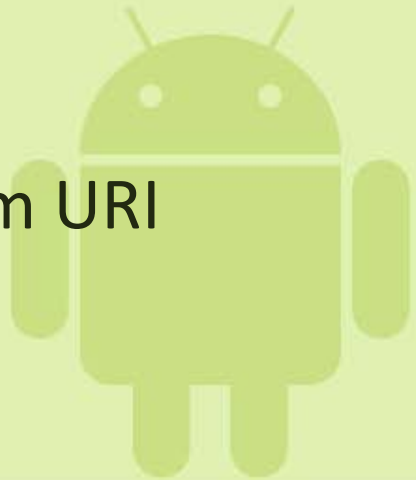
- Simplest way:

```
URL url = new URL("http://inter.net/file.ext");
InputStream is = url.openStream();
...
```

Aside – URL vs. URI

- All URLs are URIs; not all URIs are URLs.
  - URL: http://, https://, ftp://, mailto:...
  - URI **I**dentifies a resource, URL **L**ocates it.
  - For more, see: http://ajaxian.com/archives/uri-vs-url-whats-the-difference

- Three classes: URL, URI, Uri
  - URL/URI – Java standard library
  - Uri is Android specific – can convert to/from URI
  - URL has network connection methods.



Problems with Simple Approach

- No way of uploading data – one-way.
- Essentially no error handling
  - Unacceptable for a mobile internet connection.

- Solution - URLConnection

Using URLConnection

- Call URL.openConnection instead of openStream()
  - openStream() = openConnection().getInputStream()
- setConnectTimeout()
  - Sets the timeout for a connection
  - getInputStream() throws java.net.SocketTimeoutException if reached.
- getOutputStream()
  - Allows writing to the HTTP connection – two-way.

Intro of the Day – Backup Managers

- Store persistent application data to remote "cloud" storage as restore point.
  - Restored when app is reinstalled after factory reset or new Android device.
  - Completely transparent to user.

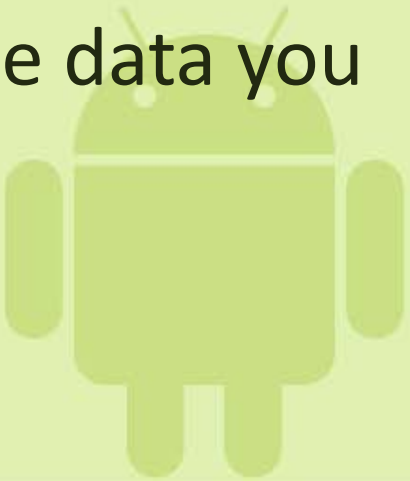
- Added in Android 2.2

Words of Caution

- Not available on all devices
  - If not, no adverse effect
- Cloud storage and transport service can differ across devices.
  - Be wary of storing usernames, passwords, and other sensitive information.

Using Backup Manager

- Declare backup agent in AndroidManifest file..
- Register with a Backup Service
  - Generally, http://code.google.com/android/backup/index.html
  - Gives an API key associated with your application.
- Extend BackupAgentHelper to specify the data you want to back up.



Declaring Backup Agent

```
<manifest ... >
    <application android:label="MyApplication"
                 android:backupAgent="MyBackupAgent"
                 android:restoreAnyVersion=“false">
        <activity ... >
            ...
        </activity>
        <meta-data
            android:name="com.google.android.backup.api_key"
            android:value="…" />
    </application>
</manifest>
```

Extending BackupAgentHelper

- Will back up:
  - SharedPreferences with SharedPreferencesBackupHelper
  - Internal storage files with FileBackupHelper

Example of BackupAgentHelper

```
public class MyBackupAgent extends BackupAgentHelper {
   static final String PREFS_BACKUP_KEY = "prefs";
   static final String FILES_BACKUP_KEY = “files";

   void onCreate() {
       SharedPreferencesBackupHelper helper =
           new SharedPreferencesBackupHelper(this,
               “prefs_file”);
       addHelper(PREFS_BACKUP_KEY, helper);
       FileBackupHelper helper2 =
           new FileBackupHelper(this, “file1”, ...);
       addHelper(FILES_BACKUP_KEY, helper2);
   }
}
```

Thread Safety

- Recall: SharedPreferences are thread-safe, so can access in application and backup simultaneously.

- Files are not!
  - Must use some kind of locking when accessing files.
  - Backup needs to use this lock too…
    - Override onBackup and onRestore in BackupAgentHelper
      - Just call superclass version inside of the lock.

More Backup Tips

- For more custom Backup, extend BackupAgent instead of BackupAgentHelper.
  - Versions of the data backup format
  - Backup portions of a large file, not all of it
  - Backup an SQLite database
- Can test backup/restore using adb
- For more, see: http://developer.android.com/guide/topics/data/backup.html

Final Week (Tentative)

- Monday – Internet II
  - Interacting with an Internet API
    - Facebook Graph: http://developers.facebook.com/docs/api
  - MapView & Location Tools
- Wednesday – Multimedia
  - Camera
  - Audio/Video, 3D Graphics
- Friday – Release & Monetization
  - Support for multiple platforms
  - Integrating advertising, releasing on Android market