Apache Wookie (Incubating)

Creating your first widget

Ross Gardler
wookie-dev@incubator.apache.org

http://incubator.apache.org/wookie

Objective focussed outline

- Creating a widget from the template
- Running in developer mode
- Modify the widget
- Using external libraries
- Using external APIs (not via the proxy)
- Setting widget properties
- Packaging for distribution

Task focussed outline

- Create a development environment
- Create a hello world widget from supplied templates
- Run Wookie in standalone mode
- Edit the widget
- Create a real widget with geo-lookup
- Display map of current location
- Store settings
- Package widget as .wgt

Development Environment

Create Development Environment

- What you need:
  - Wookie source code
  - Ant + Ivy
  - JDK
  - Your favourite editor
    - Javascript, HTML, CSS
    - Java (if you want to hack on the server)

Using the Workshop CD

- Copy the “wookie” folder to your preferred project location
- Copy “apache-ant” into your preferred applications application directory
- Install JDK from “java” folder
- Configure your environment
  - Set ANT_HOME
  - Add $ANT_HOME/bin to your $PATH

Your first widget

Hello World!

Create a Hello World Widget

- Wookie comes with a skeleton widget template:

```
cd $WOOKIE_HOME/widgets
ant seed-widget
```

- Answer some questions about your widget
  - Short Name: **part1**
  - Description: **Code from first practical session of Wookie training day**
  - Default height: **320**
  - Default width: **200**
- You now have a widget in $WOOKIE_HOME/widgets/[SHORT_NAME]

Widget File Structure

config.xml

```
- <widget id="http://wookie.apache.org/widgets/part1" version="0.1" width="200" height="320">
    <name>part1</name>
  - <description>
      Code from first practical session of Wookie training day
    </description>
    <content src="index.html"/>
    <icon src="images/icon.png"/>
    <access network="false"/>
    <author>Apache Wookie (Incubating) Team</author>
  - <licence>
      Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTIC
      file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the
      required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BA
      License for the specific language governing permissions and limitations under the License.
    </licence>
  </widget>
```

index.html

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="pragma" content="no-cache"/>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
    <title>part1</title>
  </head>
  <body>
    <h1>Hello World...</h1>
  </body>
</html>
```

Running Wookie in Developer Mode

Developer mode allows easy building and testing of widgets

Developer Mode

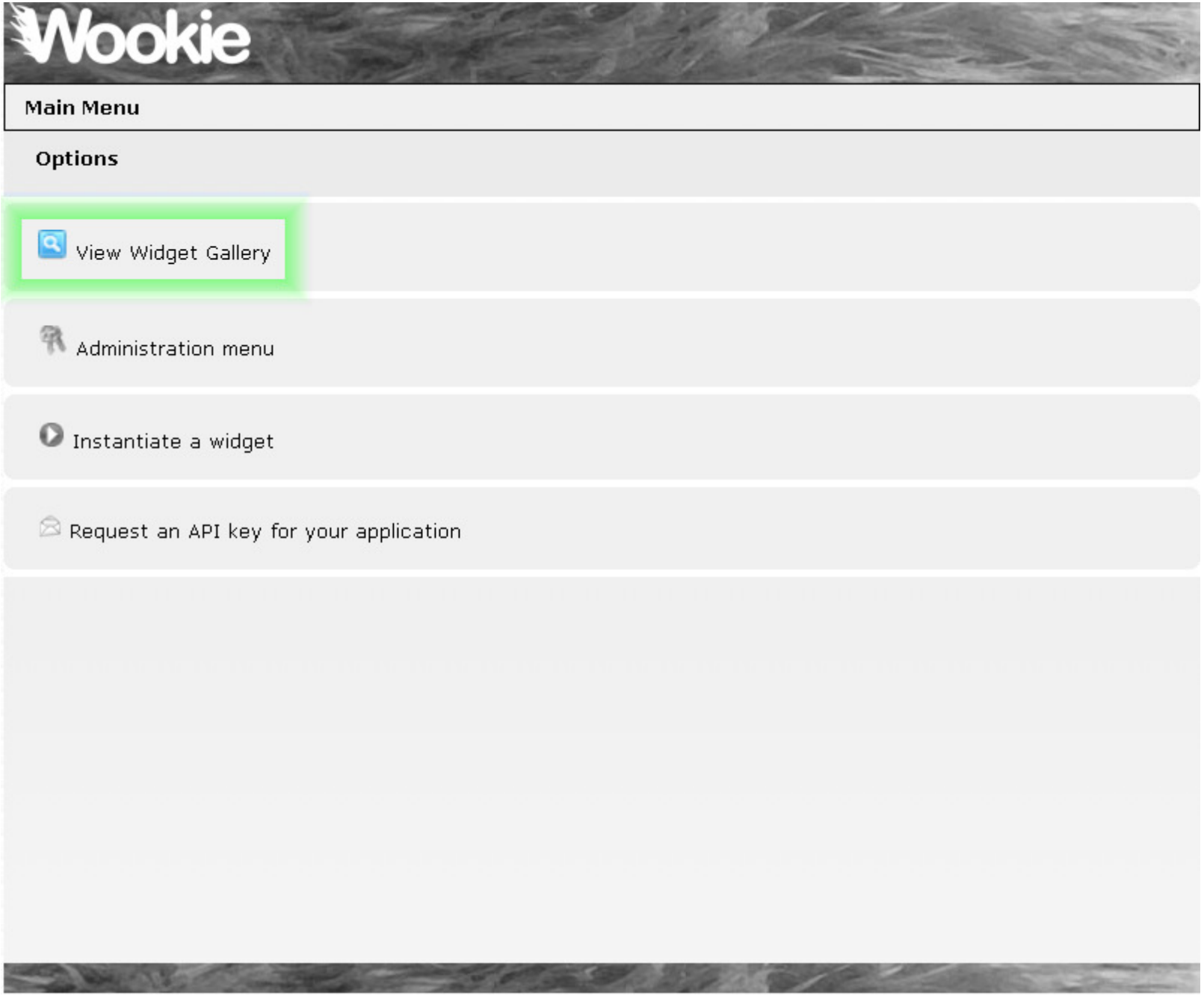
- Building Wookie auto deploys all widgets in the $WOOKIE_HOME/widgets directory
- We can run in development mode with:

```
cd ..
ant clean-db
run
```

- Visit http://localhost:8080/wookie

- [ASIDE] To run in debug mode

```
ant -Djvmargs="-Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n" run
```


Wookie
Main Menu
Options
View Widget Gallery
Administration menu
Instantiate a widget
Request an API key for your application

Wookie

menu

Current Widgets

Vote

basic voting widget

Demo

Natter

basic chat widget

Demo

API Test

A W3C API Testing Widget

Demo

Bubbles

A Bubbles game

Demo

part1

Code from first practical session of Wookie training day

Demo

Wave Test: Sudoku

Google Wave Sudoku Widget

Demo

menu

Wookie

Menu

Widget Demo

Hello World...

Menu

Editing static parts of a widget

Changing the layout of your widget

Editing a Widget

- $WOOKIE_HOME/widgets/part1/index.html
  - The initial widget page
- Edit it in your favourite editor
  - Perhaps to say **“Hello Wookie World”**
- Save

Adding some style

- It’s just HTML, so use CSS
  - style/screen.css is added by template
- Try changing the style in index.html, perhaps:

```
<span class="recent_change">Wookie</span>
```

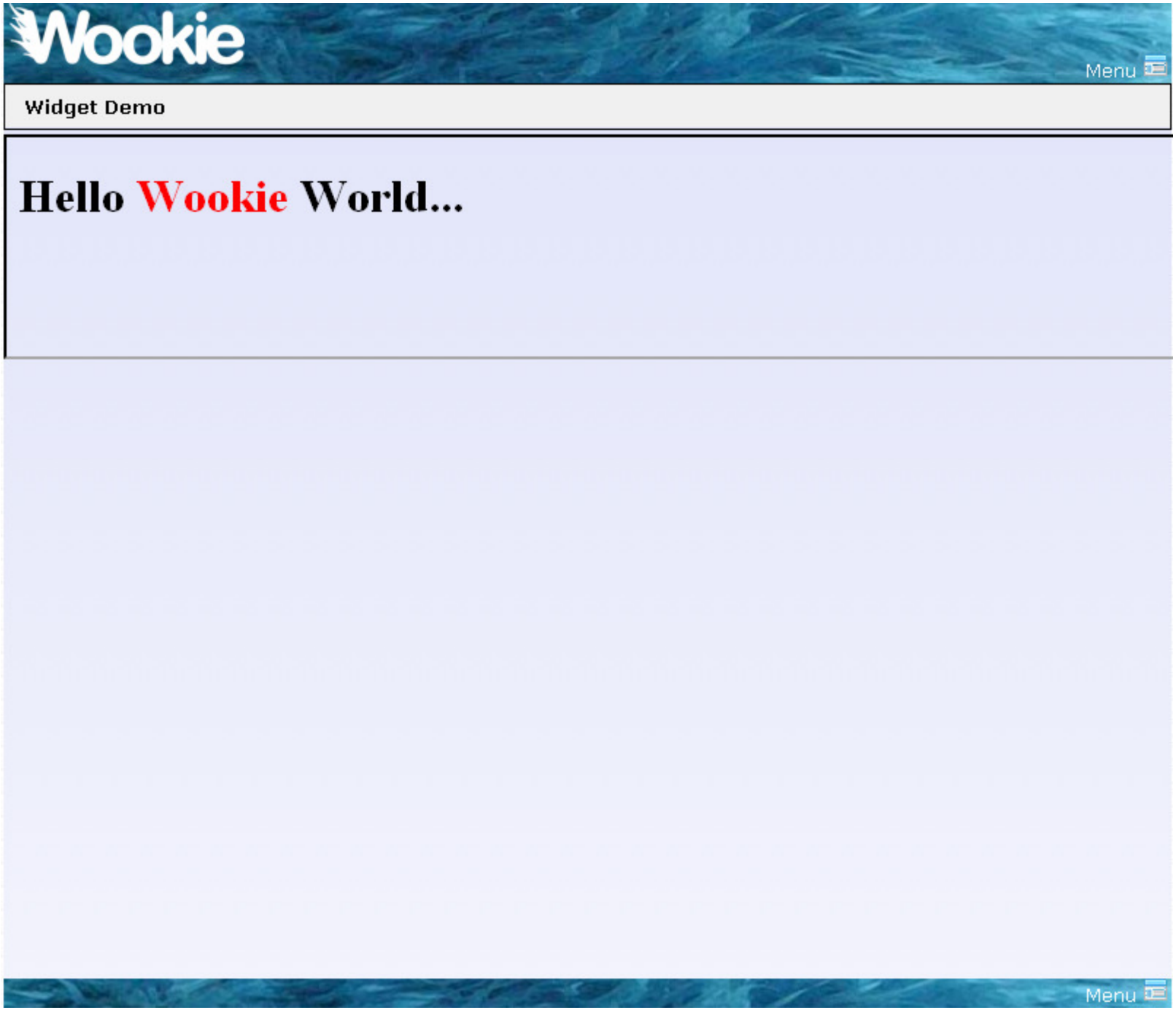
- Don’t forget the css:

```
.recent_change {
  color: red;
}
```

- Deploy (you will be asked the name of the widget to deploy)

```
ant deploy-widget
```

- Refresh Browser


Wookie
Menu
Widget Demo
Hello Wookie World...
Menu

Lets do something real...

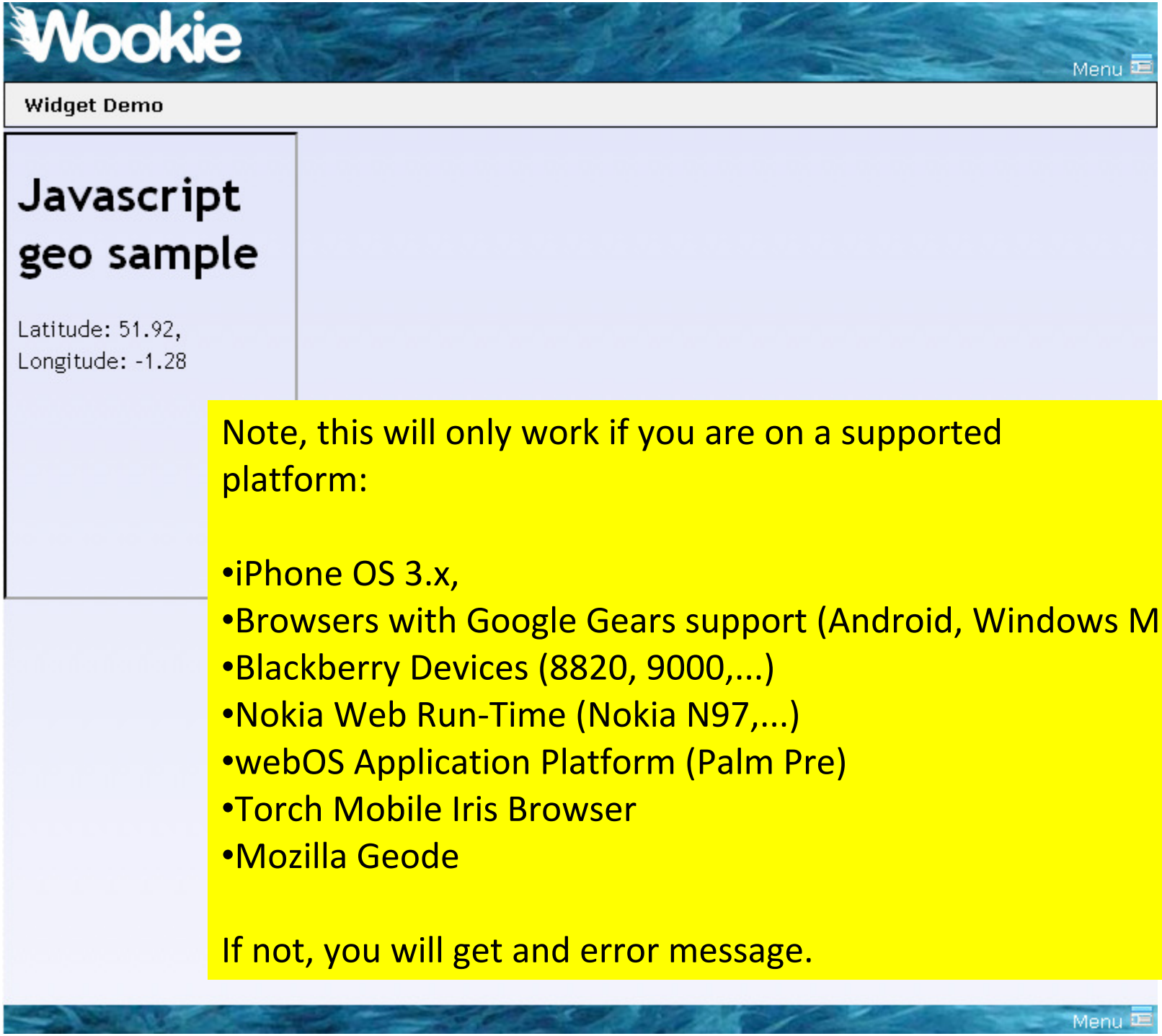
Map widget with geo-location

Geolocation Widget

- There's a great Javascript library for geolocation
  - http://code.google.com/p/geo-location-javascript
- Lets see what we can do with it
  - From $WOOKIE_HOME/tutorials/geoLocation
    - Replace contents of index.html with the contents of basicGeoWithoutFallback.html
    - Copy geo.js to the scripts directory of your widget
- Deploy

```
ant deploy-widget
```

- Refresh your browser


Wookie
Menu
Widget Demo
Javascript geo sample
Latitude: 51.92,
Longitude: -1.28
Note, this will only work if you are on a supported platform:
•iPhone OS 3.x,
•Browsers with Google Gears support (Android, Windows Mobile)
•Blackberry Devices (8820, 9000,...)
•Nokia Web Run-Time (Nokia N97,...)
•webOS Application Platform (Palm Pre)
•Torch Mobile Iris Browser
•Mozilla Geode
If not, you will get and error message.
Menu

The code

- Import Javascript libraries
  - <script src=*"scripts/geo.js" type="text/javascript"></script>*
- HTML elements to contain info
  -
  -

The code (2)

- Get current location using geo-location library
  - geo_position_js.getCurrentPosition(success_callback,error_callback,{enableHighAccuracy:true});
- Update the page using DOM
  - document.getElementById("latitude").innerHTML = p.coords.latitude.toFixed(2);
  - document.getElementById("longitude").innerHTML = p.coords.longitude.toFixed(2);

Simulating GPS

For development lets simulate a moving GPS device

Simulating Movement

- In development we can't walk around to test
  - geoLocation library allows simulation of an active GPS
  - lets simulate someone moving through oxford
- From $WOOKIE_HOME/tutorials/geoLocation
  - Copy geo_position_js_simulator.js to the scripts directory of your widget
  - We'll define a path for our "walker" in Javascript

OpenStreetMap

- Display a map from OpenStreetMap
  - http://www.openstreetmap.org
- From $WOOKIE_HOME/tutorials/geoLocation
  - Copy contents of osmWithSimulator.html to your widgets index.html
  - Copy map_osmWithSimulatorAndLayers.js to your widget script directory with the name map.js
  - Copy images/map to your widget images directory
- Deploy

```
ant deploy-widget
```

- Refresh your browser

Widget Demo

User has moved.

The code

- There is nothing special in this step that is unique to widgets
- The javascript is all from
  - The geo-location getting started guide
  - The OpenLayers API

Widget Settings

Personalising a widget

Storing Settings

- Add a setting to hide the status line
  - From $WOOKIE_HOME/tutorials/geoLocation
    - Copy contents of osmWithSimulator.html to your widgets index.html
    - Copy contents of map_osmWithSimulatorAndSettings.js to script/map.js
- Deploy

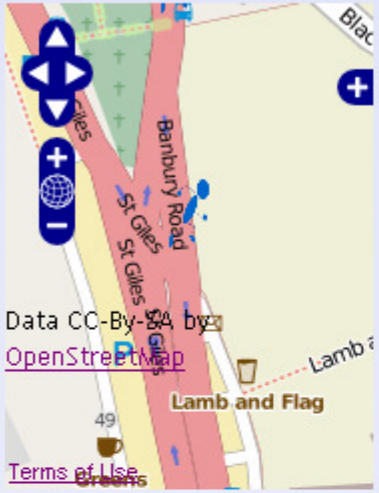
```
ant deploy-widget
```

- Refresh your browser

Wookie

Menu

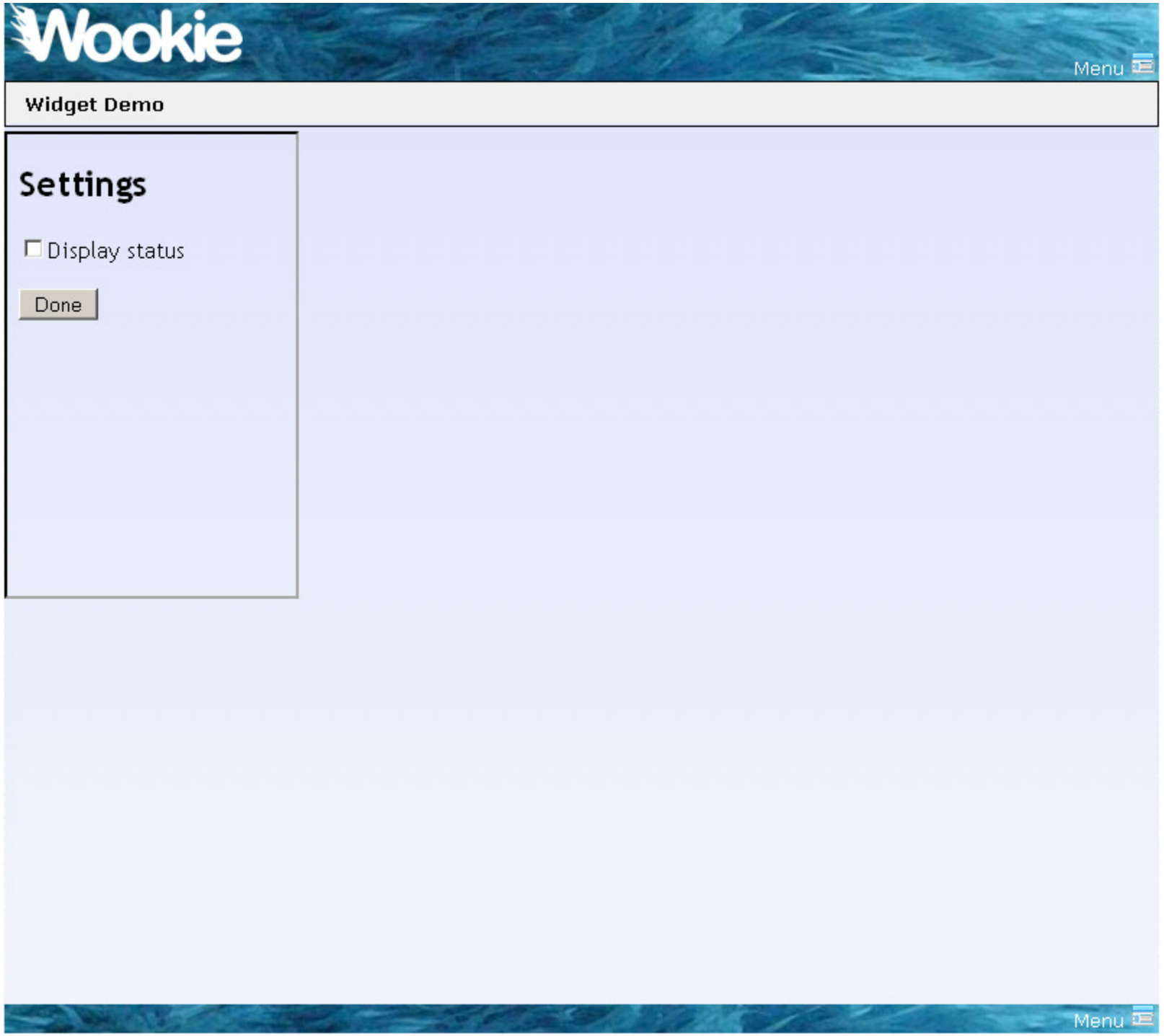
Widget Demo



User has moved.

settings

Menu


Wookie
Menu
Widget Demo
Settings
Display status
Done
Menu

The code

- Widget.preferences
  - Gives access to preferences stored for the widget
- Widget.preferences.setItem(name, value)
  - Widget.preferences.setItem('displayStatus', this.checked)
- Widget.preferences.getItem(name)
  - Widget.preferences.getItem("displayStatus")

Build for Deployment

Widgets are packaged as *.wgt files for deployment

```
oucs0040@OUCS-OSSW-RG /cygdrive/c/projects/wookie
$ ant build-widget
Buildfile: build.xml

build-widget:
    [input] What is the short name of the widget you wish to build?
part1

_get-widget-name:

_validate:

_clean:

_init:

build-widget:
     [echo] Widget package created at c:\projects\wookie\build/widgets/part1.wgt


BUILD SUCCESSFUL
Total time: 2 seconds
```

Deploy to a server

- To deploy a widget on a running server
  - Build the wgt file
  - Place the file in the “upload” directory of the server
    - webapp/wookie/upload

You’re done, you can now build widgets

Lets summarise

Widget development cycle

- Create a widget from the template
- Deploy in developer mode
  - Test
    - manually or using a framework such as Selenium
  - Edit
  - Re-Deploy to running development server
- Package for distribution