Contents

[Architecture 1](#_Toc367458753)

[Web Architecture 1](#_Toc367458754)

[Mobile Architecture 2](#_Toc367458755)

[How to Add a Page 2](#_Toc367458756)

[Build 2](#_Toc367458757)

[Web 3](#_Toc367458758)

[Localhost 3](#_Toc367458759)

[Production 3](#_Toc367458760)

[Mobile 4](#_Toc367458761)

[Localhost 4](#_Toc367458762)

[Production 4](#_Toc367458763)

[Deploy 4](#_Toc367458764)

[Web 4](#_Toc367458765)

[Localhost 4](#_Toc367458766)

[Production 4](#_Toc367458767)

[Mobile 5](#_Toc367458768)

[Localhost 5](#_Toc367458769)

[Production 5](#_Toc367458770)

[Patches 5](#_Toc367458771)

[Test 5](#_Toc367458772)

[Mobile Testing in the Browser 5](#_Toc367458773)

[Web Testing 5](#_Toc367458774)

[Beta Testing 5](#_Toc367458775)

[How To 6](#_Toc367458776)

[Add A New Page 6](#_Toc367458777)

# Architecture

## Web Architecture

Client: Angular, Twitter Bootstrap

Server: REST-based, Lift (liftweb.net)

Lift is used for its Boot.scala class, its Mapper framework and to handle REST-based url calls

LittleBlueBird/src/main/webapp/js/app.js – This is where all the routes are defined. If you want to find out where a URL takes you, start with this file.

## Mobile Architecture

Client: Angular, JQueryMobile

Server: makes the same REST-based api calls as the web version

LittleBlueBird/www – everything for the mobile app is under here. This is where phonegap expects files to be.

LittleBlueBird/www /index.html – This is the main page of the mobile app. This is a GENERATED page. This page is the combination of the pages found here: LittleBlueBird/fragments/html

Even the “fragment” pages are themselves generated. For every fragment page, you will see a page just above/below with a –nofooter.html suffix. This was a quick and dirty solution to having consistent footers on every page.

One big honkin’ index.html file, One big honkin’ controller.js…

Index.html and controllers.js are GENERATED files. They are generated by the build.xml file

Run this target in build.xml: copy

The copy target copies everything from LittleBlueBird/www to LittleBlueBird/src/main/webapp/www. Having these files under src/main/webapp lets you test the mobile app from the browser.

build.xml takes several files and combines them all in to one index.html file.

build.xml also takes several files and combines them all in to one controllers.js file

**No $location Object**

The mobile app explicitly avoids using angular’s $location object. Instead, we use JQueryMobile’s navigation paradigm: All pages are actually contained in the index.html. This file is HUGE. Pages are indentified by <div> tags via ‘id’ attributes and data-role=”page” attributes.

<div id="wishlist" data-role="page">

This is the beginning of a “page” in index.html. You would get to this page via an <a> tag with an href like this: href=”#wishlist”

### How to Add a Page

Create html file in LittleBlueBird/fragments/html

Add the file to build.xml, target: create-index

# Build

## Web

### Localhost

C:\users\bdunklau\> netstat –a –o –n

Look for the process using port 80 (top of the list)

C:\users\bdunklau\> taskkill /F /PID [pid]

Cd lit[tab]

C:\users\bdunklau\LittleBlueBird\> sbt

>container:start

Rebuilding

up arrow twice, ENTER (container:stop)

up arrow twice, ENTER (container:start)

The stop/start is all you have to do if scala files change. container:start recompiles before starting jetty.

Jetty is configured to look at the LBB project for html/js/css files.

### Production

LittleBlueBird/src/main/webapp/WEB-INF/web.xml

Delete this part if building for tomcat:

<!-- FOR JETTY ONLY - REMOVE WHEN DEPLOYING TO TOMCAT -->

<servlet>

<!-- Override init parameter to avoid nasty -->

<!-- file locking issue on windows. -->

<servlet-name>default</servlet-name>

<init-param>

<param-name>useFileMappedBuffer</param-name>

<param-value>false</param-value>

</init-param>

*</servlet>*

Start 🡪 cmd

Cd c:\users\bdunklau\LittleBlueBird

C:\users\bdunklau\littlebluebird> sbt

> package

Go to: C:\Users\bdunklau\LittleBlueBird\target\scala-2.9.1

War file: **littlebluebird\_2.9.1-1.0.war**.

Rename as: **gf.war**

Make a copy: **gf-yyyy-MM-dd.war**

web.xml – Ctrl-Z

**Now let’s deploy the war file**

Go to [www.littlebluebird.com/manager/html](http://www.littlebluebird.com/manager/html)

User: manager or admin

Pass: Pi…..

If you get page not found, you probably made your hosts file point littlebluebird.com to 127.0.0.1.

Go to c:\windows\system32\drivers\etc\hosts and comment out the line starting with 127.0.0.1

Go back to [www.littlebluebird.com/manager/html](http://www.littlebluebird.com/manager/html) do this in **Chrome** because Chrome seems to pick up changes to the hosts file – Firefox seems to cache pages

Deployment to eatj.com takes about 5 mins

## Mobile

### Localhost

Eclipse 🡪 LittleBlueBird/build.xml 🡪 “copy” target

**If scala files changed,**

Up arrow twice, ENTER (container:stop)

Up arrow twice, ENTER (container:start)

### Production

Eclipe

Right-click LittleBlueBird

Team 🡪 Commit

**Uncheck src/main/resources/props/default.props**

Click “Commit and Push”

<https://build.phonegap.com/apps/504068/builds>

Unlock the iOS and Android keys

Click “Update Code”

Need instructions for pushing new versions to app store and Google Play

# Deploy

## Web

### Localhost

C:\windows\system32\drivers\etc\hosts

Edit this file so that [www.littlebluebird.com](http://www.littlebluebird.com) is associated with 127.0.0.1

When you built for localhost, you also deployed to localhost

[www.littlebluebird.com/gf](http://www.littlebluebird.com/gf)

### Production

Use Chrome

[www.littlebluebird.com/manager/html](http://www.littlebluebird.com/manager/html)

manager/Pi…. Or admin/Pi…..

Deploy C:\Users\bdunklau\LittleBlueBird\target\scala-2.9.1\gf.war

Takes about 5 mins

## Mobile

### Localhost

[www.littlebluebird.com/gf/www](http://www.littlebluebird.com/gf/www)

This is the main page of the mobile app. It will work in a browser. The Facebook integration probably won’t work and the scanner obviously won’t work. But you can do everything else and hit the localhost db.

You cannot deploy the app to a phone and have it run against the laptop.

### Production

Production means deploy to a phone and hit the production web and db servers.

## Patches

For deploying individual files.

FileZilla 🡪 QuickConnect 🡪 [bdunklau@s156.eatj.com](mailto:bdunklau@s156.eatj.com)

Drag/drop files as needed

If scala files changed, probably best to do a “package” deployment via sbt

# Test

## Mobile Testing in the Browser

**Special “test” config:** In build.xml, see the target “create-controllers”. That target contains this:

<!-- fileset file="fragments/js/test.js" / --> <!-- not for production -->

## Web Testing

## Beta Testing

Beta testing mostly pertains to the mobile at this time (9/19/13)

**iPhone users**

Need their UDID number

Added to Adhoc provisioning file

Regenerating the Adhoc provisioning file

Email the file to them

They install it

Email them this link to download the mobile app: <https://build.phonegap.com/apps/504068/download/ios>

**Android users**

Need to be invited into the Google Community I set up

Then either look for the app in Google Play or download it from here:

<https://build.phonegap.com/apps/504068/download/android>

# How To

## Add A New Page

partials directory: put the new page here

app.js – add a new route

If anonymous access allowed, add to var nonsecure =

urlrewrite.xml: If you want F5/Refresh-ability, add entry here