[1 Architecture 1](#_Toc370480957)

[1.1 Web Architecture 1](#_Toc370480958)

[1.2 Mobile Architecture 2](#_Toc370480959)

[1.2.1 How to Add a Page 2](#_Toc370480960)

[2 Build 2](#_Toc370480961)

[2.1 Build Website 3](#_Toc370480962)

[2.1.1 Build Website for Localhost 3](#_Toc370480963)

[2.1.2 Build Website for Production 3](#_Toc370480964)

[2.2 Build Mobile App 4](#_Toc370480965)

[2.2.1 Build Mobile App for Localhost 4](#_Toc370480966)

[2.2.2 Build Mobile App for Production 4](#_Toc370480967)

[3 Deploy 4](#_Toc370480968)

[3.1 Deploy Website 4](#_Toc370480969)

[3.1.1 Deploy Website to Localhost 4](#_Toc370480970)

[3.1.2 Deploy Website to Production 4](#_Toc370480971)

[3.2 Deploy Mobile App 5](#_Toc370480972)

[3.2.1 Deploy Mobile App to Localhost 5](#_Toc370480973)

[3.2.2 Deploy Mobile App to Production 5](#_Toc370480974)

[3.3 Deploy Patches 5](#_Toc370480975)

[4 Test 5](#_Toc370480976)

[4.1 Test Mobile App in the Browser 6](#_Toc370480977)

[4.2 Test the Website 6](#_Toc370480978)

[4.3 Beta Testing 6](#_Toc370480979)

[5 How To 6](#_Toc370480980)

[5.1 Add A New Page 6](#_Toc370480981)

[6 Third Party Libraries 6](#_Toc370480982)

# 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

## Build Website

### Build Website for 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.

### Build Website for 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

## Build Mobile App

### Build Mobile App for Localhost

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

**If scala files changed,**

Up arrow twice, ENTER (container:stop)

Up arrow twice, ENTER (container:start)

### Build Mobile App for Production

Eclipse

Build.xml 🡪comment out this line like this

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

Run the default target in build.xml (“copy”)

Right-click LittleBlueBird

Team 🡪 Commit

Do not commit

* src/main/resources/props/default.props
* the private key in src/main/resources
* any .gitignore files
* any files with -2012 suffixes
* any files in src/main/webapp/www

Click “Commit and Push”

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

Unlock the iOS and Android keys

iOS key: Dist Key 2

Android key: lbb-for-android

Click “Update Code” 🡪 “Pull Latest”

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

# Deploy

## Deploy Website

### Deploy Website to 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)

### Deploy Website to Production

Prepare users for downtime:

update app\_version set downmessage = some message about going down when/how long

Assuming you have already built the war file

Assuming you have already renamed it gf.war

Assuming you have already made a dated copy of gf.war (because gf.war will always be overwritten by whatever the current release is)…

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

Drag the gf.war file from laptop to /webapps folder

Tomcat will auto-undeploy then auto-deploy

Pretty dang simple

Here’s the old way of deploying (prior to 10/25/13)

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

## Deploy Mobile App

### Deploy Mobile App to 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.

### Deploy Mobile App to Production

Build.phonegap.com

Download the .ipa, .apk, etc file

iPhone: Need mac for this. Use Application Loader

Google: (need instructions)

## Deploy 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

## Test Mobile App 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 -->

## Test the Website

## 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

# Third Party Libraries

**On the Web**

Tuckey – for url rewriting [www.tuckey.org](http://www.tuckey.org)

Version: ?

Key files: urlrewrite.xml

Bootstrap

Version: 2.0.2

Docs: <http://bootstrapdocs.com/>

Docs: <http://bootstrapdocs.com/v2.0.2/docs/>

Angular

Version: ?

Lift – [www.liftweb.net](http://www.liftweb.net)

Version: ?

For: Boot.scala, mapper classes, REST service

**Mobile**

JQueryMobile

Version: ?

For: client-side GUI

Angular

Version: ?

Support for this doc…

Good article on TOC numbering: <http://shaunakelly.com/word/numbering/numbering20072010.html>