



ATLAS

AngularJS Development Environment

View Slides: nbcuots.github.io/atlas

Download Slides: nbcuots.github.io/atlas/atlas.pdf

Install Atlas github.com/scottnath/atlas

Learn About Atlas scottnath.github.io/atlas

A Tale of Two Systems

**Client-Side
Application**



**Development
Environment**



AngularJS Application



Why AngularJS over other js frameworks?

- ✓ Massive community
- ✓ Open source *and* backed by Google
- ✓ Custom HTML Elements
- ✓ Complete client-side solution
- ✓ Built with CRUD applications in mind
- ✓ Two-way data binding
- ✓ Testable

What is AngularJS?

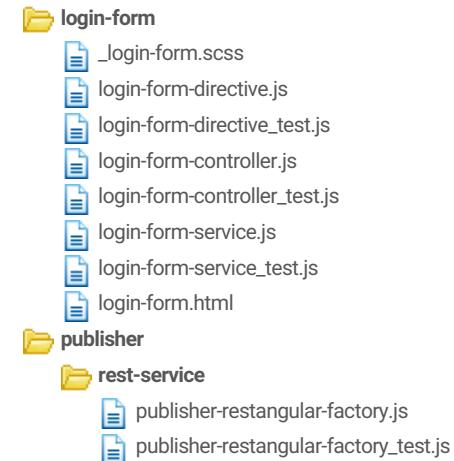
A Javascript MVC framework, similar to Ember.js or Backbone.js. ✓ Dependency Injection

A Folder structure

Atlas is following [Google's AngularJS Application Structure Best Practice Recommendations](#).

That means:

- ✓ Recursive hierarchical structure
- ✓ Component's parts grouped together
- ✓ Pieces split into properly named files
- ✓ Descriptive file names
- ✓ Tests are packaged with the code they are testing



- Break down code into **smallest pieces**
- Small pieces combined = **re-usable pattern groups**
- Re-usable patterns stacked = **Page/App**



Smallest piece



Complete pattern



Complete project



Modularity Example: Login Form

login-form

- _login-form.scss
- login-form-directive.js
- login-form-directive_test.js
- login-form-controller.js
- login-form-controller_test.js
- login-form-service.js
- login-form-service_test.js
- login-form.html

Sign In

SSO ID
206123456

Password [Forgot your Password?](#)

••••••••••

Remember my ID



Modularity: Open Source

AngularJS' Dependency Injection allows us to use third-party components.

Example:

AngularJS AutoFields (7.1kb)

```
{  
  property: 'sso',  
  type: 'text',  
  attr: {  
    pattern: '^[0-9]*$',  
    maxlength: 9,  
    minlength: 9,  
    required: true  
  },  
  msgs: {  
    minlength: 'SSO is 9 letters',  
    pattern: 'Only Numbers in SSO',  
    required: 'SSO required'  
  },  
  validate: false,  
  ngMessages: true  
}
```

```
<input type="text" id="sso" name="sso" minlength="9"  
      maxlength="9" ng-model="data.sso"  
      pattern="^[0-9]*$" placeholder="Type SSO"  
      ng-init="data.sso='206123456'" required />  
<div ng-messages="myForm2.sso.$error">  
  <div ng-message="pattern">Only Numbers in SSO</div>  
  <div ng-message="minlength">SSO is 9 letters</div>  
  <div ng-message="required">SSO required</div>  
</div>
```



Modularity Example: Form Element

The Smallest Piece:
A Single Form Element



SSO ID
206123456

```
<label for="sso">SSO ID</label>
<input type="text" id="sso" name="sso" minlength="9"
maxlength="9" ng-model="data.sso"
pattern="^[0-9]*$" placeholder="Type SSO"
ng-init="data.sso='206123456'" required />
<div ng-messages="myForm2.sso.$error">
  <div ng-message="pattern">Only Numbers in SSO</div>
  <div ng-message="minlength">SSO is 9 letters</div>
  <div ng-message="maxlength">SSO is 9 letters</div>
  <div ng-message="required">SSO required</div>
</div>
```

Make it a pattern then... **reuse it everywhere**

1. Pattern -> AngularJS directive
2. Add to our Pattern Library
3. Import, use with *any* project

```
<sso-text-input></sso-text-input>
```



Modularity Example: Login Form

Pattern Group:
An SSO Login Form



Sign In

SSO ID
206123456

Password [Forgot your Password?](#)

Remember my ID **Sign In**

```
<form name="ssoLogin" class="ng-pristine ng-valid">
  <h1>Sign In</h1>
  <sso-text-input></sso-text-input>
  <login--password></login--password>
  <login--rememberid></login--rememberid>
  <button>Sign In</button>
</form>
```

Make it a pattern then... reuse it *everywhere*

1. Pattern -> AngularJS directive
2. Add to our Pattern Library
3. Import, use with *any* project

```
<sso-login-form></sso-login-form>
```



Modularity Example: Build a Page

Patterns on Patterns:
Time to Build a Page!



NBCUniversal

Sign In

SSO ID
206100581

Password

Forgot your Password?

Remember my ID

Sign In

```
...
<nbc-security-header></nbc-security-header>
<sso-login-form></sso-login-form>
<nbc-security-links></nbc-security-links>
<nbc-copyright-footer></nbc-copyright-footer>
...
```

Make it a pattern then... reuse it *everywhere*

1. Pattern -> AngularJS directive
2. Add to our Pattern Library
3. Import, use with *any* project

```
<nbc-login-page></nbc-login-page>
```

Modularity: Patterns Everywhere



S 12 / Ep 8
A mid-season twist could bring an eliminated chef back, then the judges shop for the Elimination Challenge.

Bravo



Amazing Fact
Could Admiring her really
happened? What would life on a
spaceship really be like? Find out.

SYFY



▶
Sarinda Swan
Get the scoop on Graceland

USA

```
<article>
  <figure>
    
    <figcaption>Full Episode</figcaption>
  </figure>
  <div>
    <h3>S 12 / Ep 8</h3>
    <p>A mid-season twist could bring an eliminated chef back, then the judges shop for the Elimination Challenge.</p>
  </div>
</article>
```

```
<article>
  <image-promo-video></image-promo-video>
  <text-promo-video></text-promo-video>
</article>
```

```
<article-promo-video></article-promo-video>
```

- ✓ Dependency injection
= *stackable, modularized functionality*
- ✓ Built with testability in mind
= *consistent code that's proven to be functional*
- ✓ Massive community
= *free work & tons of solutions*
- ✓ Efficient and fast for end-users
= *happy customers*
- ✓ Can be used on non-AngularJS projects
= *can roll out new pieces on existing sites*

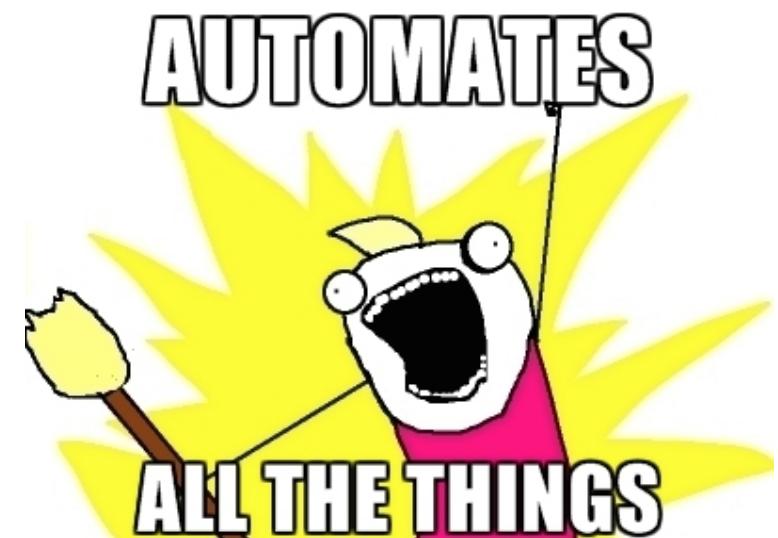
Atlas Development Environment



Atlas: What it does

it's a...

- ✓ code checking
- ✓ file gathering
- ✓ css compiling
- ✓ distribution building
- ✓ web server
that does
- ✓ unit testing
and
- ✓ end to end testing.





Package Managers

Packages = Software

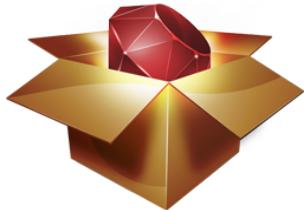
Package Management = Installs / upgrades / configures / removes software

Advantage = All developers using SAME software

Examples in other tech:

- Java: Maven
- Apple: Homebrew
- PHP: Composer
- Linux: apt-get, yum

Package Managers in Atlas



Ruby Gems

- ✓ Installs Ruby packages
- ✓ Cuts back on root installs

Examples: SASS, Compass, Singularity



Node Package Manager

- ✓ Installs NodeJS packages
- ✓ Small development scripts

Examples: Uglify, GH Pages, Chalk



Bower

- ✓ Client-Side Tech
- ✓ Grabs both public *and* private repos
- ✓ NBCUOTS pattern library

Examples: AngularJS, AutoFields, jQuery



The taskrunner automates our development process by running various tasks...but what are those tasks?



GulpJS

- ✓ Code Checking
- ✓ File building
- ✓ Minification
- ✓ Concatination
- ✓ Distribution Creation
- ✓ List goes on...

BrowserSync

- ✓ Auto-refresh connected browsers
- ✓ Only refreshes changed files
- ✓ Syncs all browsers including clicks and scrolls

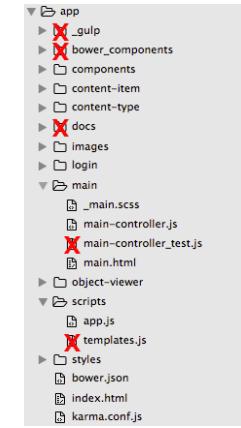


File Watching

Gulp watches our files for changes.

We can choose specific files, or whole groups with wildcards

```
scripts: [ // Application javascripts
  '!app/docs/**', // ignore documentation in app
  '!app/bower_components/**', // ignore bower-ingested scripts
  '!app/**/*_test.js', // ignore our test scripts
  '!app/scripts/templates.js', // template script
  '!app/_gulp/**', // ignore all files in _gulp directory
  'app/**/*.js'], // main application file
```



What is globbing?

1. Looks through directories for a given file type
2. Compiles a list of all matching files
3. Writes that list to a file

```
// cssGlobbingBegin  
// found .scss files @import-ed here  
// cssGlobbingEnd
```

```
// cssGlobbingBegin  
@import '../components/directive-example/_directive-example.scss';  
@import '../components/gallery/_gallery.scss';  
@import '../components/login-form/_login-form.scss';  
@import '../components/show/_show.scss';  
@import '../content-item/_content-item.scss';  
@import '../content-type/_content-type.scss';  
@import '../login/_login.scss';  
@import '../main/_main.scss';  
@import '../object-viewer/_object-viewer.scss';  
// cssGlobbingEnd
```

What is linting?

1. Inspects Code
2. Finds common mistakes
3. Exports results to console
4. Callbacks allow actions

```
...
"no-undef": 1,
"no-new": 1,
"no-extra-semi": 1,
"no-debugger": 2,
"no-caller": 1,
"semi": 1,
"quotes": [1, "single"], // single quotes ONLY
"no-unreachable":2,
...
...
```

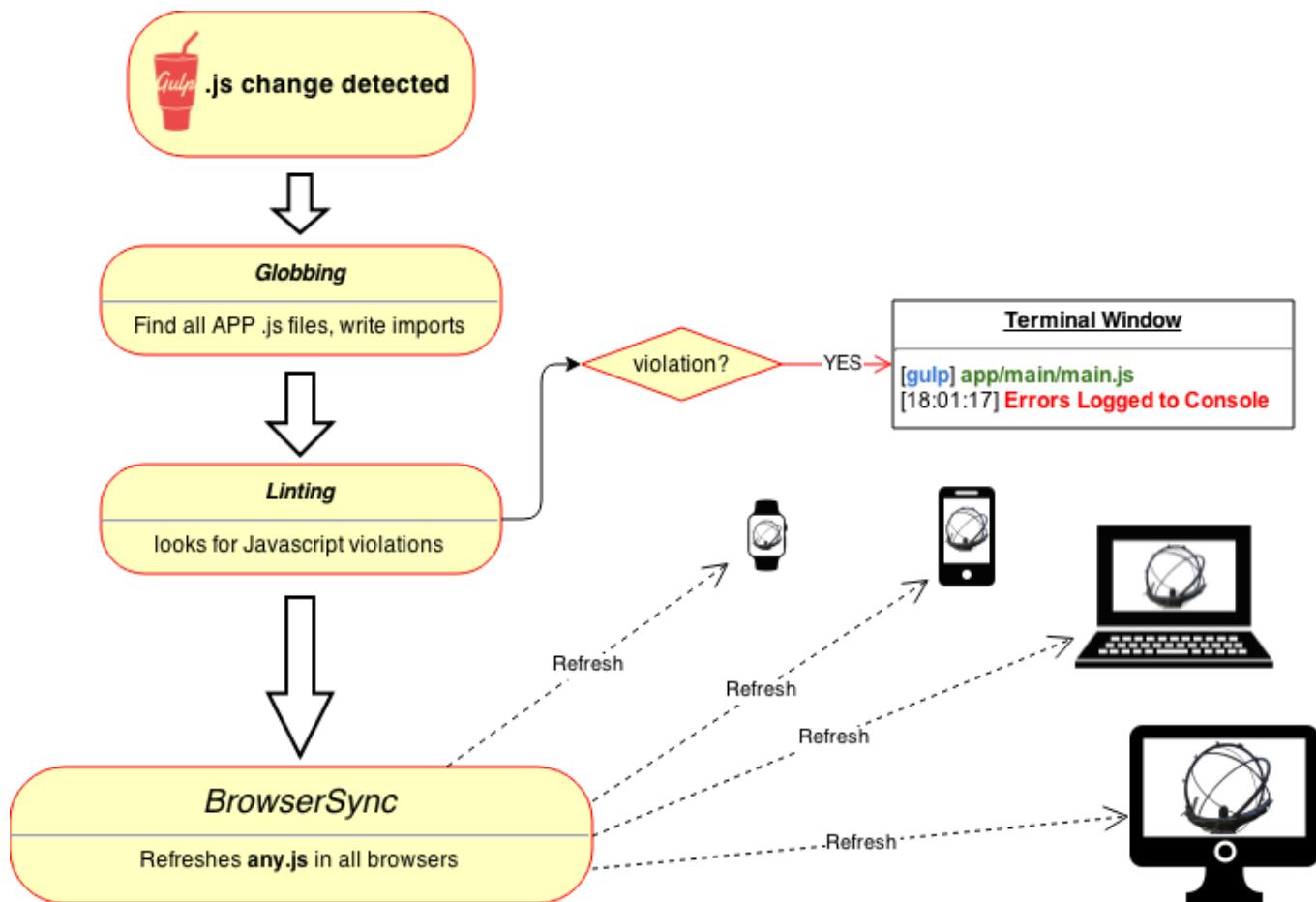
```
$scope.scopedObject = [
{
  'key': "bravo", // BAD: using double quotes
  'title': 'Bravo TV: Watch What Happens',
  'url': 'http://bravotv.com',
  'description': 'The Real Housewives of New York City, The Singles Project ...',
  'an array': ['arrayItem1','arrayItem2','arrayItem1'],
  'an object': {'objectKey1':'objectValue1','objectKey2':'objectValue2'}
},
...
...
```

```
...
[12:41:01] Starting 'dev:eslint'...
----- DEVELOPMENT: ESLint Javascript Linting
[12:41:01] gulp-inject 22 files into index.html
[BS] 1 file changed (index.html)
[12:41:01] Finished 'dev:js:globbing' after 122 ms
[12:41:01]
/Users/scottnath/development/gitrepos/atlas-proposed/app/object-viewer/object-viewer-controller.js
  16:15  warning  Strings must use singlequote  quotes

✖ 1 problem

[BS] 1 file changed (object-viewer-controller.js)
...
```

Javascript Flowchart





- Write less code
- Reusable Functions
- Does all compilation to CSS

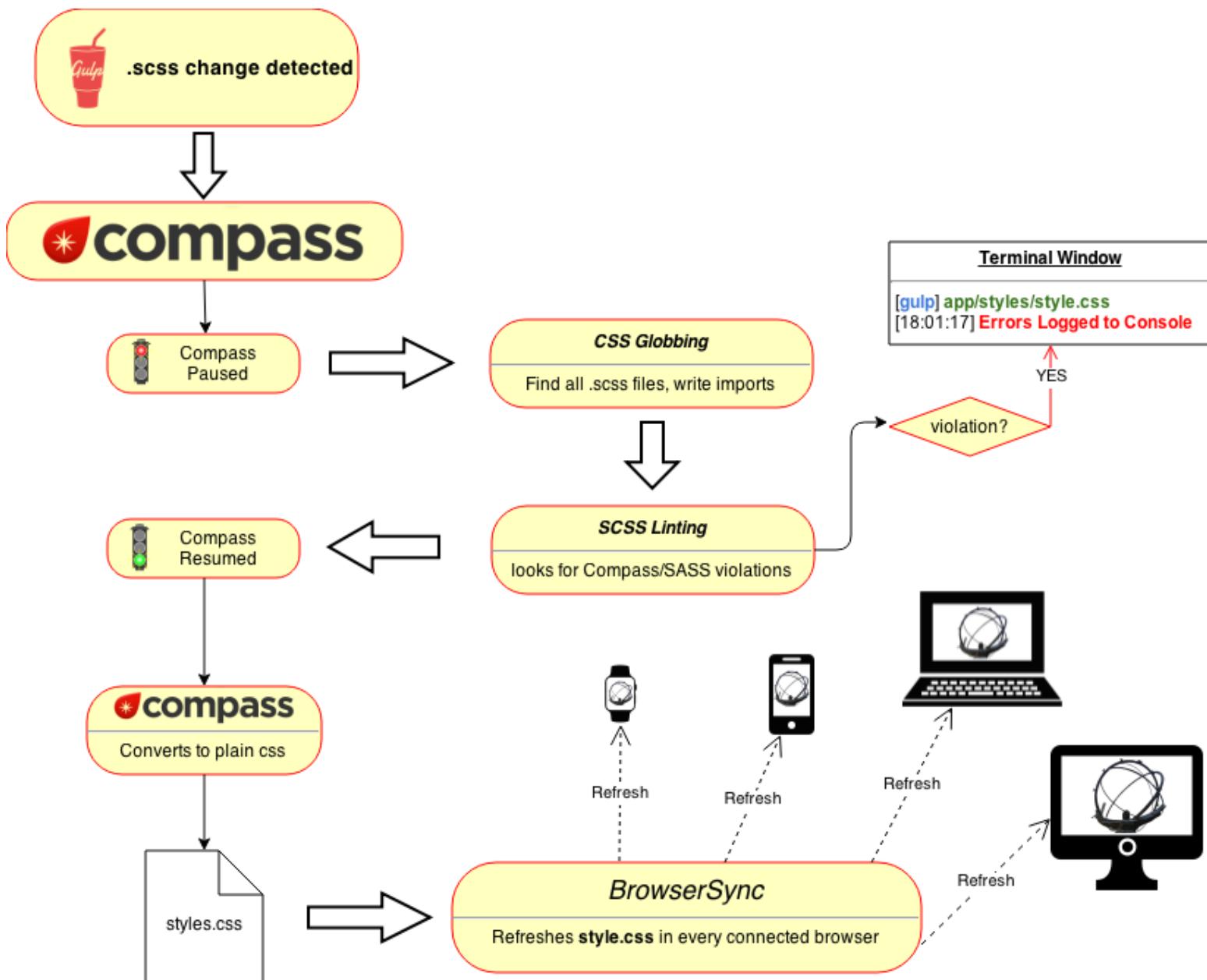
Before Compass Compilation

```
// Box shadow with custom settings
#box-shadow-custom {
  @include box-shadow(red 2px 2px 10px);
}
```

After Compass Compilation

```
#box-shadow-custom {
  -moz-box-shadow: red 2px 2px 10px;
  -webkit-box-shadow: red 2px 2px 10px;
  box-shadow: red 2px 2px 10px;
}
```

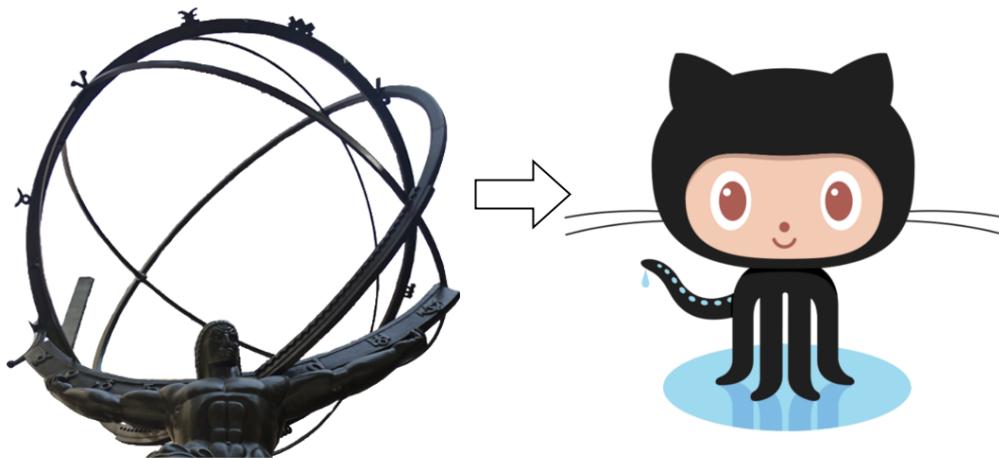
CSS Flowchart



Application Distribution

Atlas creates a distribution version of our application which:

- ✓ Combines and Minifies
- ✓ Revisions
- ✓ Makes a compact and deliverable system
- ✓ Delivers our system via GitHub

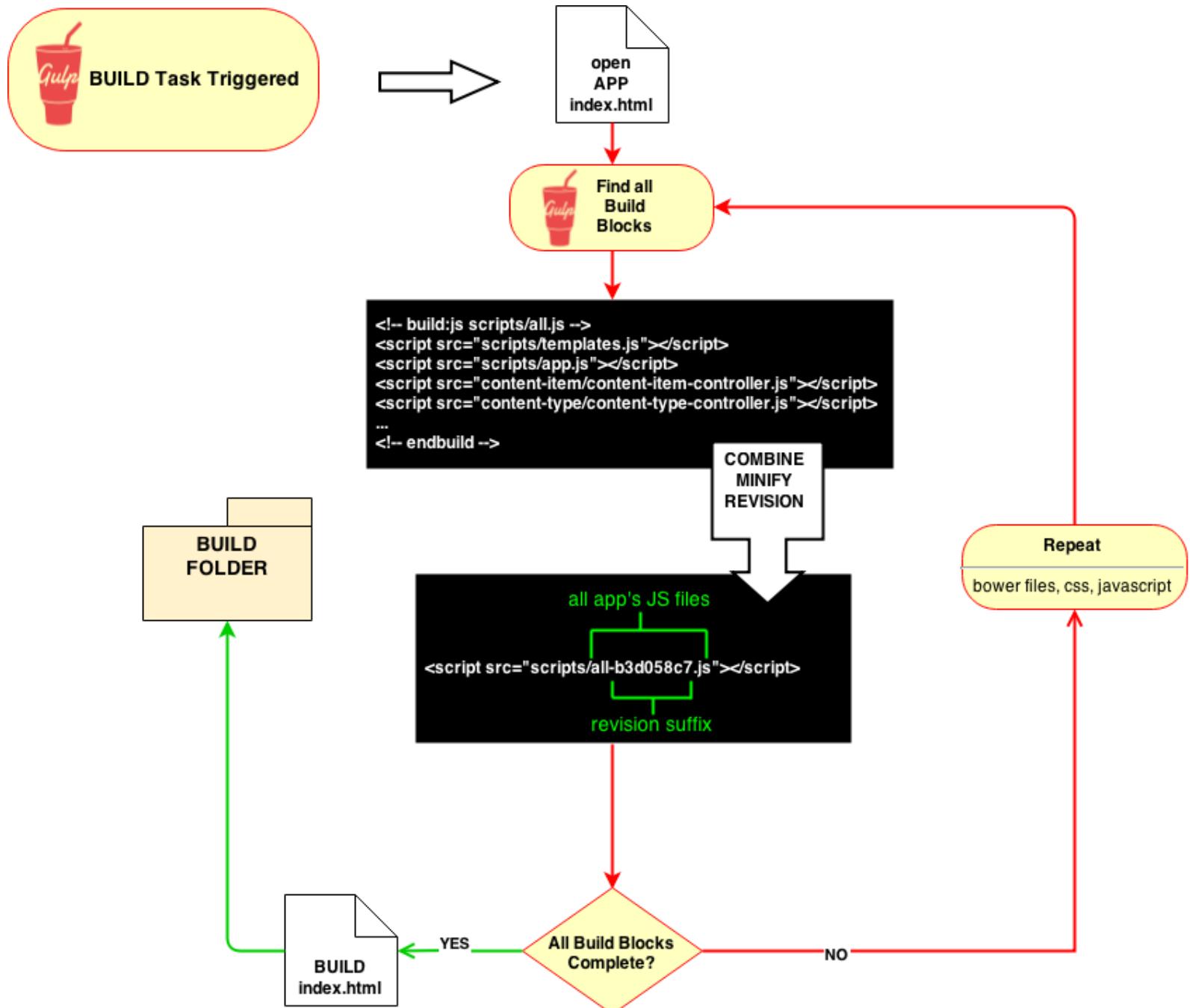


Concatenation, Minification, and Revisioning

```
<!-- build:js bower/bower.js -->
<!-- bower:js -->
<script src="angular/angular.js"></script>
<script src="angular-cookies/angular-cookies.js"></script>
<script src="angular-sanitize/angular-sanitize.js"></script>
<script src="angular-route/angular-route.js"></script>
<script src="angular-touch/angular-touch.js"></script>
<script src="angular-animate/angular-animate.js"></script>
<script src="lodash/dist/lodash.compat.js"></script>
<script src="restangular/dist/restangular.js"></script>
<script src="angular-object-viewer/object-viewer.js"></script>
<script src="marked/lib/marked.js"></script>
<script src="angular-marked/angular-marked.js"></script>
<script src="angular-mocks/angular-mocks.js"></script>
<script src="angular-aria/angular-aria.js"></script>
<script src="hammerjs/hammer.js"></script>
<script src="angular-material/angular-material.js"></script>
<!-- endbower -->
<!-- endbuild -->
<!-- build:js scripts/all.js -->
<!-- inject:js -->
<script src="scripts/templates.js"></script>
<script src="scripts/app.js"></script>
<script src="karma.conf.js"></script>
<script src="content-item/content-item-controller.js"></script>
<script src="content-type/content-type-controller.js"></script>
<script src="login/login-controller.js"></script>
<script src="main/main-controller.js"></script>
<script src="gallery/galleries-directive.js"></script>
<script src="gallery/galleries-factory.js"></script>
<script src="gallery/galleries-small-directive.js"></script>
<script src="gallery/gallery-controller.js"></script>
<script src="gallery/gallery-directive.js"></script>
<script src="gallery/gallery-small-directive.js"></script>
<script src="login-form/login-form-directive.js"></script>
<script src="show/show-directive.js"></script>
<script src="show/show-factory.js"></script>
<script src="show/show-small-directive.js"></script>
<script src="show/shows-directive.js"></script>
<script src="yes-no-filter/yes-no-filter.js"></script>
<!-- endinject -->
```

```
<script src="bower/bower-89bd38b1.js"></script>
<script src="scripts/all-c3c06c31.js"></script>
```

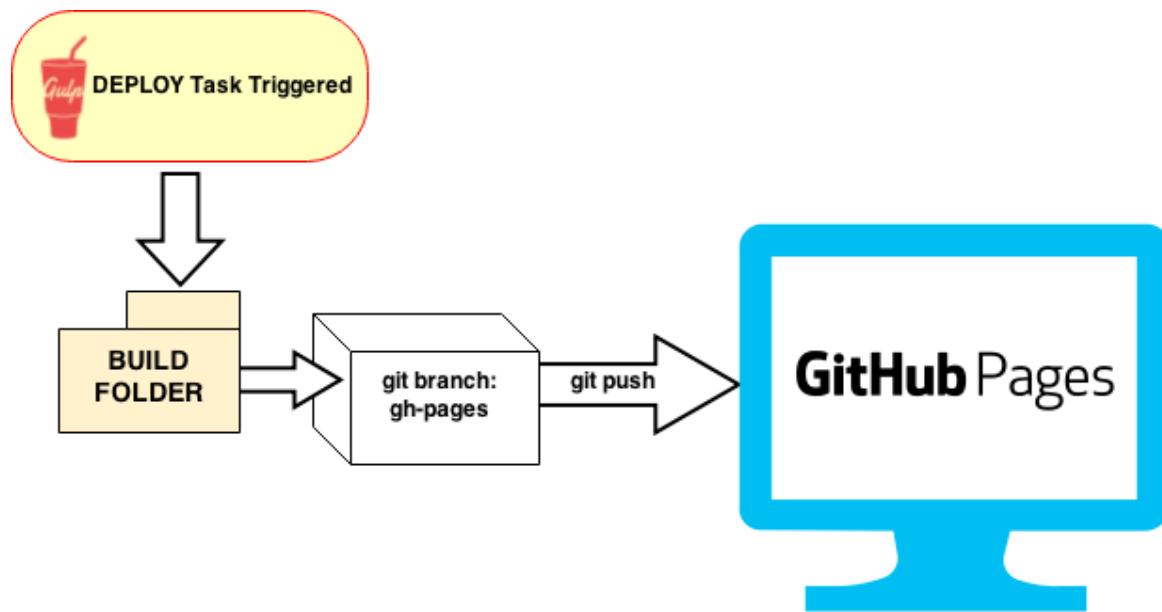
Distribution Flowchart



Build Deployment

- ✓ Deploy to any branch on any repo
- ✓ If a branch doesn't exist, creates it (on remote too)
- ✓ Virtually any actions git can do, Atlas can do

```
$ gulp deploy:ghPages
```



- ✓ Removes Setup Headaches
= *common environment for all - without all the pain*
- ✓ Package Managers
= *pull code from everywhere and everyone*
- ✓ Taskrunner
= *runs the system, so developers can just develop*
- ✓ Built In Server
= *check across all your devices instantly*
- ✓ Code Inspection
= *enforces consistent, bug-free products*
- ✓ Distribution
= *creates and deploys production-ready systems*

What's complete?

What's Complete: AngularJS

- ✓ Folder Structure (Modularity)
 - ✓ Import Public AngularJS Modules
 - ✓ Import Private AngularJS Modules
 - ✓ AngularJS Base Generator
-

What's Complete: Atlas

- ✓ Fully-Functioning Development Environment Generator
- ✓ Package Managers
- ✓ Taskrunner
- ✓ Development Server
- ✓ Globbing/Linting
- ✓ Compass/SASS
- ✓ Distribution System
- ✓ Deployment to GitHub Pages

What's left to do?

AngularJS

- Pattern Library
- REST Library Research
- SEO
- E2E/Unit Testing
- Complete File Generators
- Angular 1.4 (Spring 2015)

Atlas

- Automated Docs Generation
- Automatic Testing
- Image Minification
- Incorporate PatternLab-Like Features
- Gulp 4.0 (March 2015)

Action

Install Now

github.com/scottnath/atlas

Expanded Information

(note: tl;dr ahead)

scottnath.github.io/atlas



ATLAS

AngularJS Development Environment

View Slides: nbcuots.github.io/atlas

Download Slides: nbcuots.github.io/atlas/atlas.pdf

Install Atlas github.com/scottnath/atlas

Learn About Atlas scottnath.github.io/atlas