



Creating a Production Build Pipeline



John Papa

@john_papa | www.johnpapa.net



Build Pipeline



Gather the assets in the index.html

Concatenate them into distributable files

Copy to the build folder

Update index.html to point to the new files

Optimization

Add template caching
to the Inject task

Create an Optimize
task to gather the files

Write the “built”
index.html file

```
<!-- build:js js/lib.js -->
<!-- bower:js -->
<script src="/bower_components/jquery/dist/jquery.js">
<script src="/bower_components/angular/angular.js">
<script src="/bower_components/angular-sanitize/angu
<script src="/bower_components/bootstrap/dist/js/boo
<script src="/bower_components/extras.angular.plus/n
<script src="/bower_components/moment/moment.js"></s
<script src="/bower_components/angular-ui-router/rel
<script src="/bower_components/toastr/toastr.js"></s
<script src="/bower_components/angular-animate/angul
<!-- endbower -->
<!-- endbuild -->
```

Dev-time index.html

```
<!-- build:js js/app.js -->
<!-- inject:js -->
<script src="/src/client/app/app.module.js"></script>
<script src="/src/client/app/core/core.module.js"></
<script src="/src/client/app/customers/customers.mod
<script src="/src/client/app/dashboard/dashboard.mod
<script src="/src/client/app/layout/layout.module.js
<script src="/src/client/app/widgets/widgets.module.
<script src="/src/client/app/blocks/logger/logger.mo
<script src="/src/client/app/blocks/exception/except
<script src="/src/client/app/blocks/router/router.mo
<script src="/src/client/app/core/config.js"></scrip
<script src="/src/client/app/core/constants.js"></sc
<script src="/src/client/app/core/dataservice.js"></
<script src="/src/client/app/customers/customer-deta
<script src="/src/client/app/customers/customer-deta
```

Build-time index.html

```
<script src="js/lib.js"></script>

<script src="js/app.js"></script>
```

Gathering Assets

gulp-useref

Parses HTML comments

Similar to gulp-inject

Concatenates by default

Replaces script tags

New optimized files

<https://github.com/jonkemp/gulp-useref>

```
<!-- build:css styles/lib.css -->
<!-- bower:css -->
<!-- endbower -->
<!-- endbuild -->
```

Optimized file

```
<!-- build:css styles/app.css-->
<!-- inject:css -->
<!-- endinject -->
<!-- endbuild -->
```

Optimized file

```
</head>
```

```
<body>
```

```
<div ng-include="'app/layout/shell.html'"></div>
```

```
<!-- build:js js/lib.js -->
<!-- bower:js -->
<!-- endbower -->
<!-- endbuild -->
```

Optimized file

```
<!-- build:js js/app.js-->
<!-- inject:js -->
<!-- endinject -->
```

Optimized file

```
<!-- inject:templates:js -->
<!-- endinject -->

<!-- endbuild -->
```

The gulp-userref API

`.pipe($.userref.assets())`

Gathers assets from the HTML
comments

`.pipe($.userref.assets.restore())`

Restore the files to the
stream. index.html, for
example

`.pipe($.userref ())`

Concatenate files

Optimizing the Build

index.html

Prepare optimized index.html

Injection

Parse comments in HTML

Assets

Gather, concatenate, replace

