



A Modernized Developer's Workflow with Domino/ XPages

Eric McCormick



MWLUG 2016

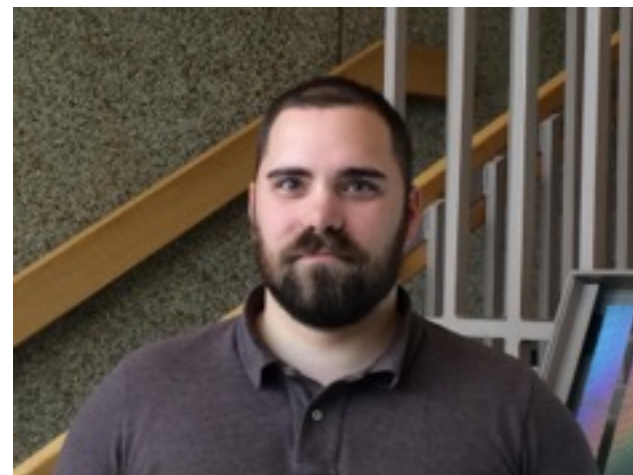
Four Seasons, Austin TX
August 17-19, 2016
Defining the Art of Collaboration

Audience



Who Am I?

- developer
- husband
- father
- veteran
- former EMT
- blogger
- ...more than I have space for



IBM CHAMPION 

[Eric McCormick .io](http://ericmccormick.io)

 [@edm00se](https://twitter.com/edm00se)

blog: edm00se.io



Who Am I?

Web Developer at **BOLDT** | BEYOND 125 YEARS

- a fourth-generation, family-owned professional construction services firm, in operation since 1889

my focus usually revolves around:

- document control
- productivity and performance metrics
- system integration
- process automation



Agenda

- What and Why
- How? (in the App)
 - Split front-end and back-end (RESTful APIs)
 - HttpServlet
 - OSGi plugin (JAX-RS)
 - XPage w/ `xe:customRestService` (Java or SSJS)
 - XAgent
 - Back-end primary logic
 - Front-end heavy app
- Observations of a corporate developer
- How? (Outside the App)
- Demos



What and Why?

- Modern web apps
 - using all the fancy new toys
- existing Domino server
 - XPages runtime
 - all the Java server stack power
 - XPages components
 - user / security model
- performance optimizations
 - lots of development tooling
- eases concerns of:
 - testing
 - documentation
 - deployment
- eases sanity



App Concerns: Splitting the App Logic

where does the application logic lie?

- business logic
- workflow logic

server-side logic:

primary logic

- enforceable rules
- notifications
- (additional) server events

client-side logic:

- how the “form” operates
- specifics to UI/UX, or the device (e.g.- camera, barcode scanner)



App Concerns: Back-End

- “best way” of establishing RESTful (or other) APIs:
 - OSGi plugin driven
 - NSF driven
 - HTTPServlets (rather, extending DesignerFacesServlet)
 - xe:restService
 - xe:customServiceBean
 - other (even Domino’s SSJS)
 - XAgent
- adding tooling:
 - OSGi plugins mean we can use “real” Eclipse for development
 - can use things like JAX-RS (which the rest of the JEE world uses!)
 - for those willing/able to make an edit to the Domino servers java.policy file:
 - GSON
 - otherwise the com.ibm.commons.util.io.json way



App Concerns: Front-End Concepts

- you can use all the fancy toys
- you “deploy” your built app to your ODP’s WebContent/ path
- properly structured, you can use back-end mocks for DDE-less preview and edits
- some tooling (angular-cli, webpack, etc.) automates this





App Concerns: Front-End Implementation

- use all the things!!1
- build for deployment
- profit

caveats:

- know what you're doing
- don't obliterate WebContent/WEB-INF/



App Concerns: Front-End Deployment

Files changed (1)

+4 -1 package.json

package.json Coverage error Side-by-side diff View file Comment ...

...

8 8

9 9

10 10

11 -

11 +

12 +

13 +

12 14

13 15

14 16

"lint": "tslint \"src/**/*.ts\"",
"test": "ng test",
"pree2e": "webdriver-manager update",
- "e2e": "protractor"
+ "e2e": "protractor",
+ "clean": "del 'ODP/WebContent/*' '!ODP/WebContent/WEB-INF'",
+ "build": "npm run clean && ng build -prod -o ODP/WebContent/"
},
"private": true,
"dependencies": {

...

21 23

22 24

23 25

26 +

24 27

25 28

26 29

"@angular/platform-browser-dynamic": "2.0.0-rc.5",
"@angular/router": "3.0.0-rc.1",
"core-js": "^2.4.0",
+ "del-cli": "^0.2.0",
"moment": "^2.14.1",
"ng2-bootstrap": "^1.0.24",
"reflect-metadata": "0.1.3",

...



Observations of This Corporate Developer

- scm for everything
- if you spend on a task, automate it
 - to the extent available
- build scripts should be:
 - minimal effort (one-line commands are easy)
 - configured as non-invasively as possible
- “anyone” should be able to pull the latest:
 - code
 - build
 - submit a feature branch / PR
- CI / CD considerations:
 - headless DDE builds
 - more?



Why a “Developer’s Workflow?”

- Automation is King
- Focus on the task at hand
- If the steps are reproducible, they should be automated/ scripted/etc.
- You can waste your time configuring and setting up
- Or you can get to work





CommitStrip.com



Beyond the App

- GitLab has CI capabilities baked in
- running GitLab in a docker container makes upgrades amazingly simple
- installing the GitLab CI runner on Windows easy
 - much easier than the setup required for [setting up Jenkins for headless DDE builds](#) (a la Cameron Gregor)
 - can set PowerShell as the default shell environment
 - the runner is registered w/ the main GitLab env, can “tag” (e.g.- Windows, DDE, PS)
 - can limit # of simultaneous executions (aka- only one DDE build at a time)
- [the possibilities are endless](#)



MWLUG 2016

Four Seasons, Austin TX
August 17-19, 2016
Defining the Art of Collaboration

Demos



Choose Your Own Adventure!

Back-End

- [generator-xsp](#)
- [HTTPServlets](#),
[xe:restService](#)

Front-End

- [generator-webapp](#)
- [generator-angular](#)
- [TypeScript](#) setup
(simplified, DIY)
- [angular-cli](#) (Angular 2)
- [vue-cli](#) ([templates](#), w/
[browserify](#) or [webpack](#))



Scaffolding an App's ODP (generator-xsp)

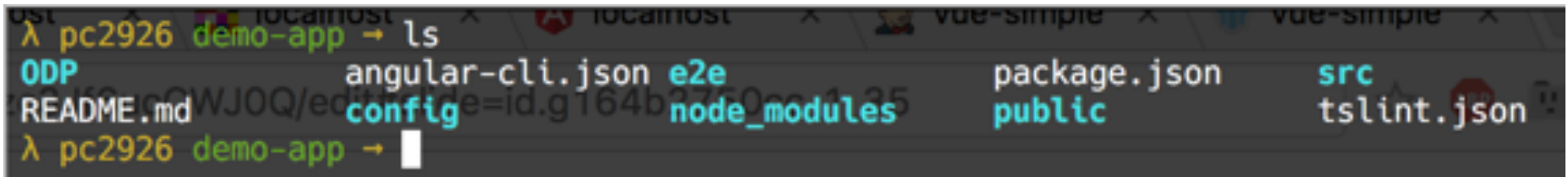
```
pc2926 demo-app → yo xsp
https://docs.google.com/presentation/d/1LgGsdZJV-aV7unNDOurYh6RxbnB4jhzc3Jf6ucCWJ0Q/edit#slide=id.g164b2750cc_1_19
Welcome to the good generator-some-app-2 generator!
Modernized Developer's Workflow
Edit View Insert Slide Format Arrange Tools Table Help All changes saved in Drive
Background... Layout... Theme... Transition...
16 What shall we call your app? demo
create bower.json
create package.json
create .bowerrc
create ODP/AppProperties/xspdesign.properties
create ODP/Code/dbscript.lsd
create ODP/Resources/AboutDocument
create ODP/Resources/Themes/app.theme
create ODP/Resources/Us lngDocument
create ODP/Views/(untitled).view
create ODP/WebContent/WEB-INF/faces-config.xml
create ODP/WebContent/WEB-INF/xsp.properties
create ODP/XPages/index.xsp
create ODP/XPages/index.xsp.metadata
create .gitattributes
create ODP/.project
create ODP/AppProperties/database.properties
create ODP/plugin.xml
create ODP/Resources/IconNote
I'm all done. Running npm install && bower install for you to install the required dependencies. If this fails, try running the command yourself.
19 npm WARN demo@ No description
npm WARN demo@ No repository field.
npm WARN demo@ No license field.
λ pc2926 demo-app → ls
ODP bower.json node_modules package.json
```



Scaffolding an App's Front-End 1 / 2

```
λ pc2926 demo-app → ls
ODP package.json
λ pc2926 demo-app → ng init
installing ng2
? Overwrite package.json? Yes, overwrite
create .editorconfig
create README.md
create src/app/app.component.css
create src/app/app.component.html
create src/app/app.component.spec.ts
create src/app/app.component.ts
create src/app/app.module.ts
create src/app/environments/environment.dev.ts
create src/app/environments/environment.prod.ts
create src/app/environments/environment.ts
create src/app/index.ts
create src/app/shared/index.ts
create src/favicon.ico
create src/index.html
create src/main.ts
create src/polyfills.ts
create src/test.ts
create src/tsconfig.json
create src/typings.d.ts
create angular-cli.json
create config/karma.conf.js
create config/protractor.conf.js
create e2e/app.e2e-spec.ts
create e2e/app.po.ts
create e2e/tsconfig.json
create .gitignore
overwrite package.json
create public/.npmignore
create tslint.json
Installing packages for tooling via npm
```

Scaffolding an App's Front-End 2/2



- change the dev/build scripts to:
 - output to ODP/WebContent/
 - not obliterate your ODP/WebContent/WEB-INF/





MWLUG 2016

Four Seasons, Austin TX
August 17-19, 2016
Defining the Art of Collaboration

Q&A



Coming to my blog soon...

- more on generator-xsp
- extending beyond my task runner series
 - to include use of webpack w/ front-end apps on Domino/XPages



MWLUG 2016

Four Seasons, Austin TX
August 17-19, 2016
Defining the Art of Collaboration

Thank You!