

DEV-1545

Beyond Domino Designer

Connect
2017

IBM

Please note

IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

P A R E N T A L

A D V I S O R Y

E X P L I C I T C O N T E N T

Lots of naked code included!

Agenda

- What, Why, How
- The tools
- Developer Workflow
- Demo
- Q & A



What, Why, How

What

Deliver modern looking web & mobile applications, that are:

- Well tested
- Familiar to use
- Frequently updated
- Maintainable by
“typical web developers”

The screenshot shows a modern web application dashboard. On the left is a blue sidebar menu with the 'CoreNG' logo at the top. The menu items include Dashboard, Components, Forms, Layout, Tables, Charts, Maps, and a purchase theme option. The main content area has a header 'Dashboard' and a search bar with 'John Smith'. Below the header are several colored cards: pink (Comments: 28), green (Likes: 831), red (Sign Ups: 204), and dark blue (Sales: 85). A large chart titled 'Server Load' shows a blue area plot with data points from 10:24:50 to 10:25:40. The chart has a note: 'Click and drag in the plot area to zoom in'. Below the chart is a section titled 'Angular 2 Contributors' with a table:

Username	Location	Bio	Email
vsavkin	Toronto	Victor toys with eclectic programming technologies and obsesses over fonts and keyboard layouts. He also makes Angular.	No Public Email

On the right side, there is a 'Recent Commits' sidebar with a list of commits from users tbosch, vici, jasonaden, and IgorMinar, each with a small profile picture and commit details.

Domino Designer

One ring to rule them all?

Why
2/2/17



If you are a Hobbit - maybe

JasmineJS
Angular
ReactNative
Selenium
ES2016
ES2015
TypeScript
Webpack
Angular2
React
CD
JavaScript
Cordova
Jenkins
NodeJS
git
npm
Ionic
Gulp
Yeoman
Travis
Grunt
CI
TDD
http/2

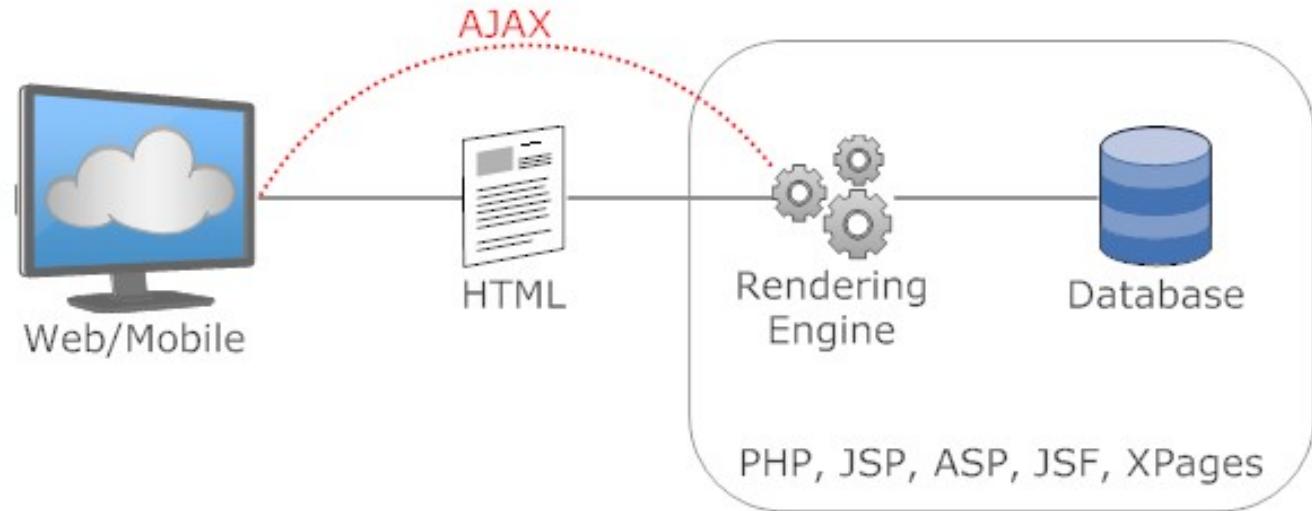
Why

Ideas



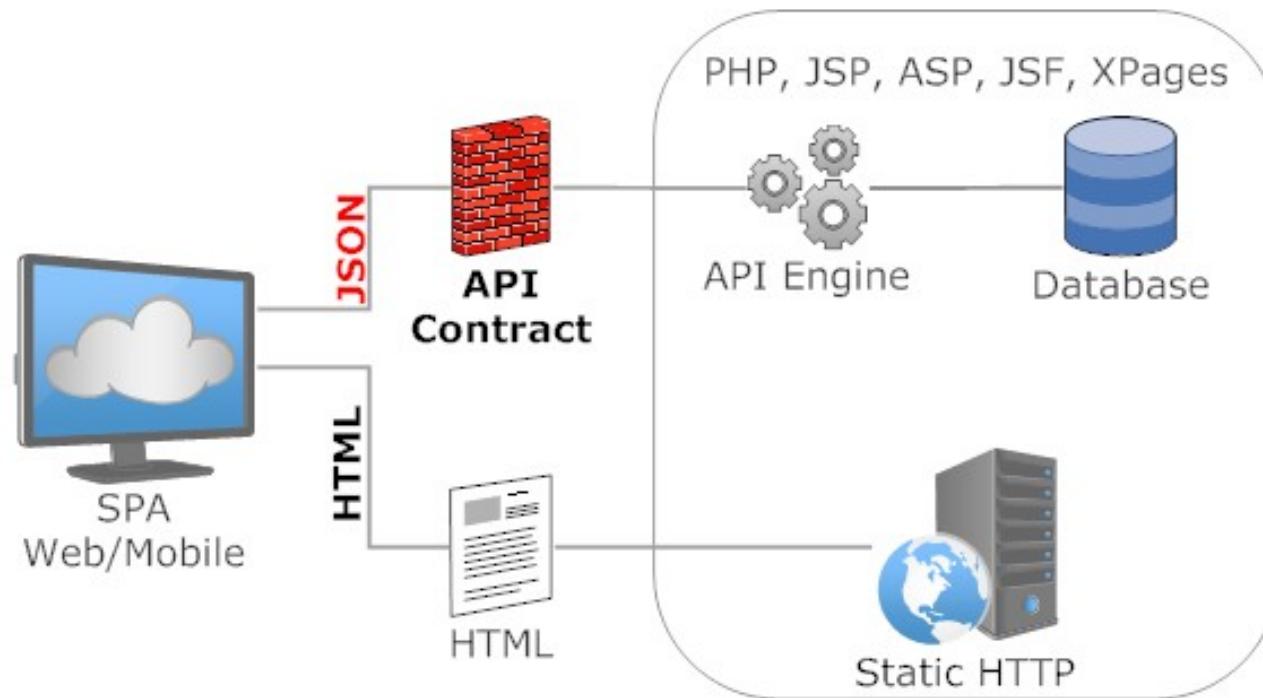
What we have been doing

"Classic" Server Side Rendering



The current fashion

"Single Page App" Rendering



Rethink development language

Domino

- Forms
- Views
- Actions
- Agents

List of
Documents in
a database

Single Page Apps

- Routes
- Services
- Views

UI rendering
as result of
a route

Don't confuse
Domino views and
SPA views!



HOW

The tools

Developer Toolbox vs. Domino Designer



Source: <http://www.strictlytoolboxes.com/>



Source: <https://www.knifecenter.com/>

And there came along....

```
stw at Stephans-MacBook-Pro-3 in ~/Code/github/Connect2017-DEV1545 on master [!?]
```

Install

node.js

<https://nodejs.org>
(currently 6.9.4 LTS)
... and ...
make friends with

npm

we take it from there

How command line irony

What – JavaScript tooling

- NodeJS
- Webpack
- Wallaby
- Angular
- Npm / yarn
- Postman





A word on tooling

By Konrad Adenauer
German Chancellor
1949 – 1963

“Was interessiert
mich mein
Geschwätz von
gestern”

Source: By Bundesarchiv,
B 145 Bild-F078072-0004
Katherine Young
CC-BY-SA 3.0, CC BY-SA 3.0 de
<https://commons.wikimedia.org/w/index.php?curid=5356485>

Update speed of core and packages



vs.

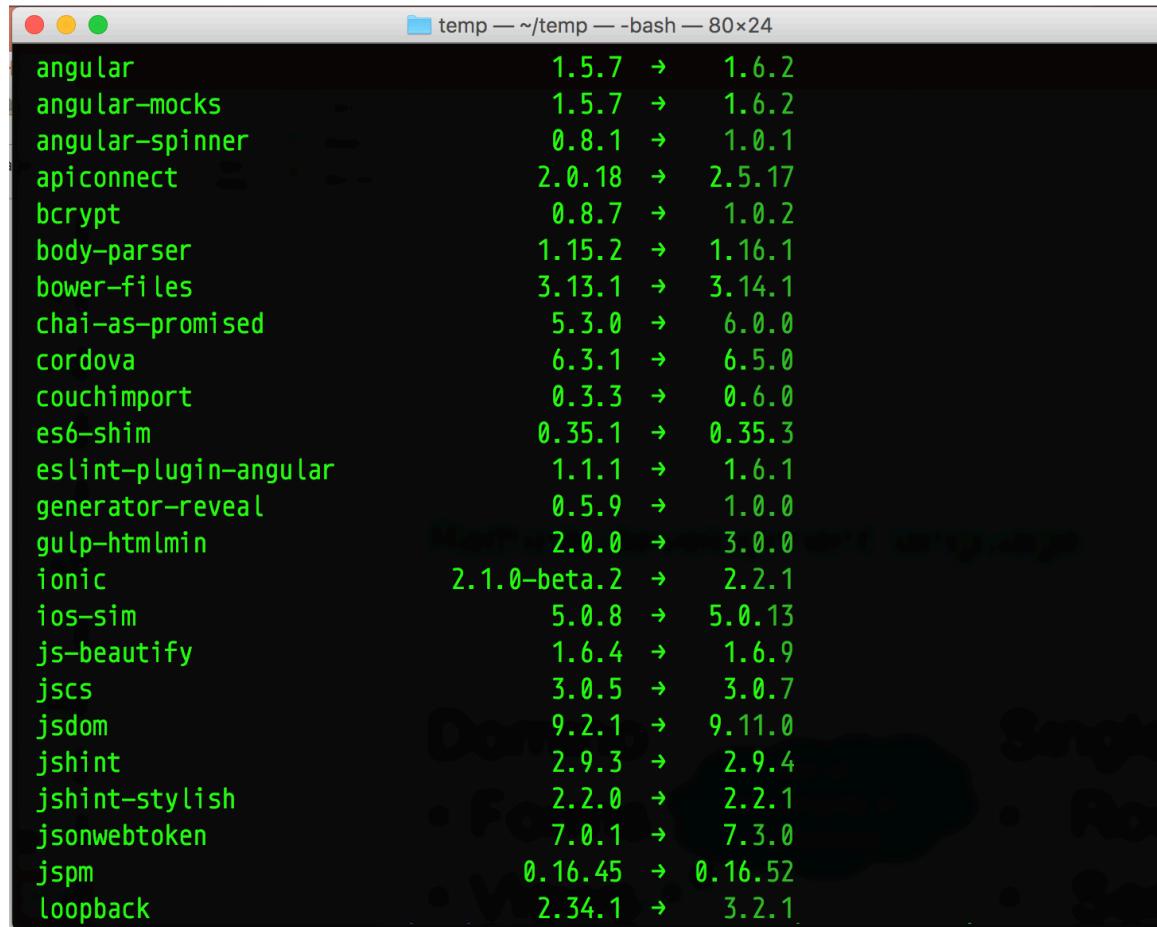


© Disney Inc

© Dreamworks Studios

npm install -g npm-check-updates

ncu -g



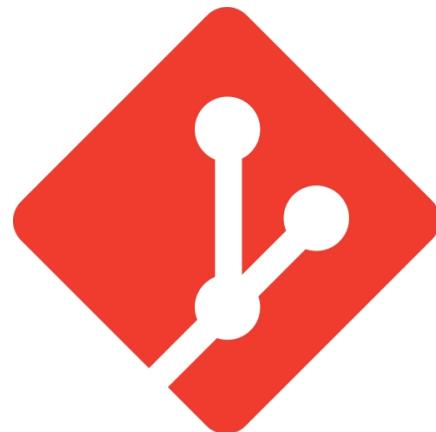
A screenshot of a terminal window titled "temp — ~/temp — bash — 80x24". The window displays a list of npm packages and their current versions followed by an arrow and the recommended update version. The packages listed include angular, angular-mocks, angular-spinner, apiconnect, bcrypt, body-parser, bower-files, chai-as-promised, cordova, couchimport, es6-shim, eslint-plugin-angular, generator-reveal, gulp-htmlmin, ionic, ios-sim, js-beautify, jscs, jsdom, jshint, jshint-stylish, jsonwebtoken, jspm, and loopback.

Package	Current Version	Recommended Version
angular	1.5.7	→ 1.6.2
angular-mocks	1.5.7	→ 1.6.2
angular-spinner	0.8.1	→ 1.0.1
apiconnect	2.0.18	→ 2.5.17
bcrypt	0.8.7	→ 1.0.2
body-parser	1.15.2	→ 1.16.1
bower-files	3.13.1	→ 3.14.1
chai-as-promised	5.3.0	→ 6.0.0
cordova	6.3.1	→ 6.5.0
couchimport	0.3.3	→ 0.6.0
es6-shim	0.35.1	→ 0.35.3
eslint-plugin-angular	1.1.1	→ 1.6.1
generator-reveal	0.5.9	→ 1.0.0
gulp-htmlmin	2.0.0	→ 3.0.0
ionic	2.1.0-beta.2	→ 2.2.1
ios-sim	5.0.8	→ 5.0.13
js-beautify	1.6.4	→ 1.6.9
jscs	3.0.5	→ 3.0.7
jsdom	9.2.1	→ 9.11.0
jshint	2.9.3	→ 2.9.4
jshint-stylish	2.2.0	→ 2.2.1
jsonwebtoken	7.0.1	→ 7.3.0
jspm	0.16.45	→ 0.16.52
loopback	2.34.1	→ 3.2.1

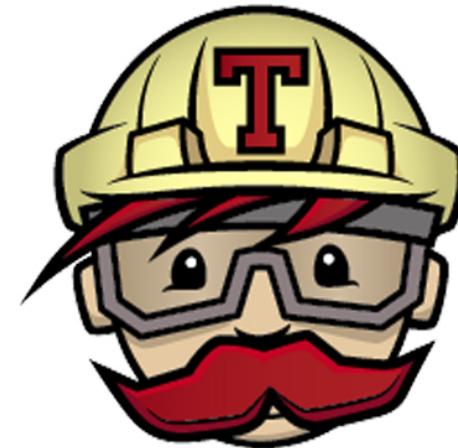
Developer workflow

What – Development flow

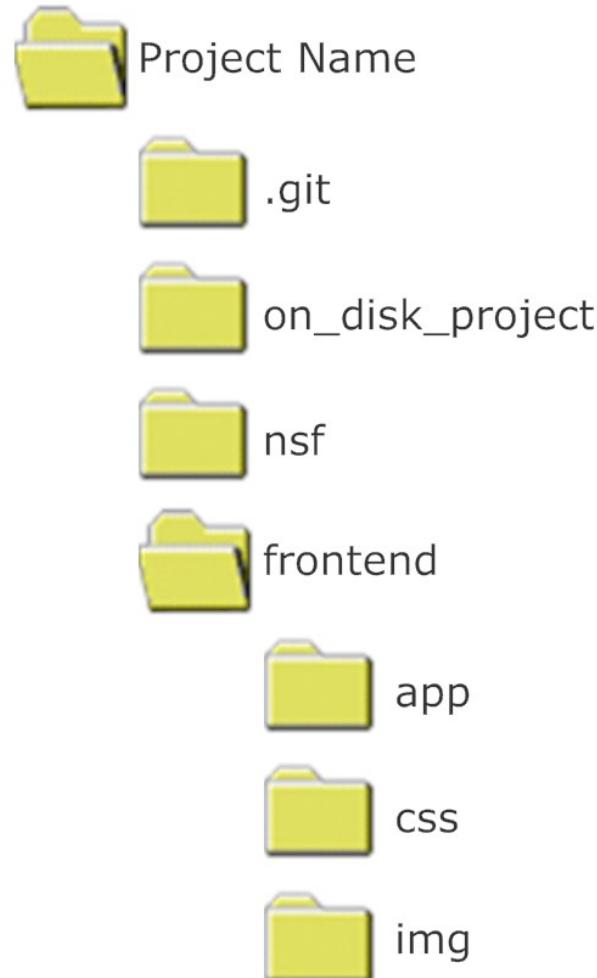
- Project Structure
- Version Control
- Testing
- Lots of testing



Jasmine



Project structure – single .git



.gitignore

```
.DS_Store  
*.nsf  
Desktop.ini  
.Spotlight-V100  
.Trashes  
Thumbs.db
```

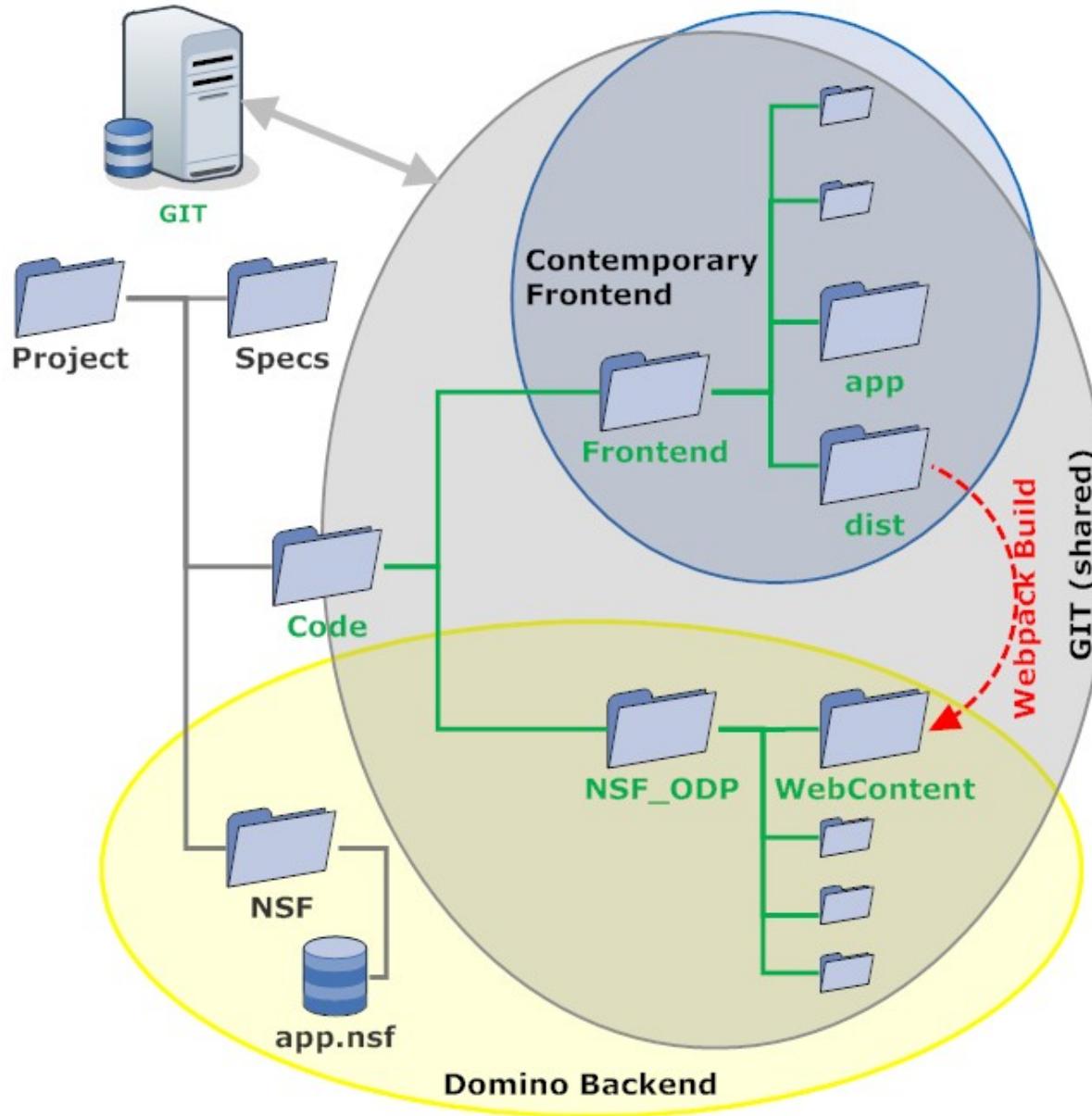
You might want to separate the repositories for front and back end



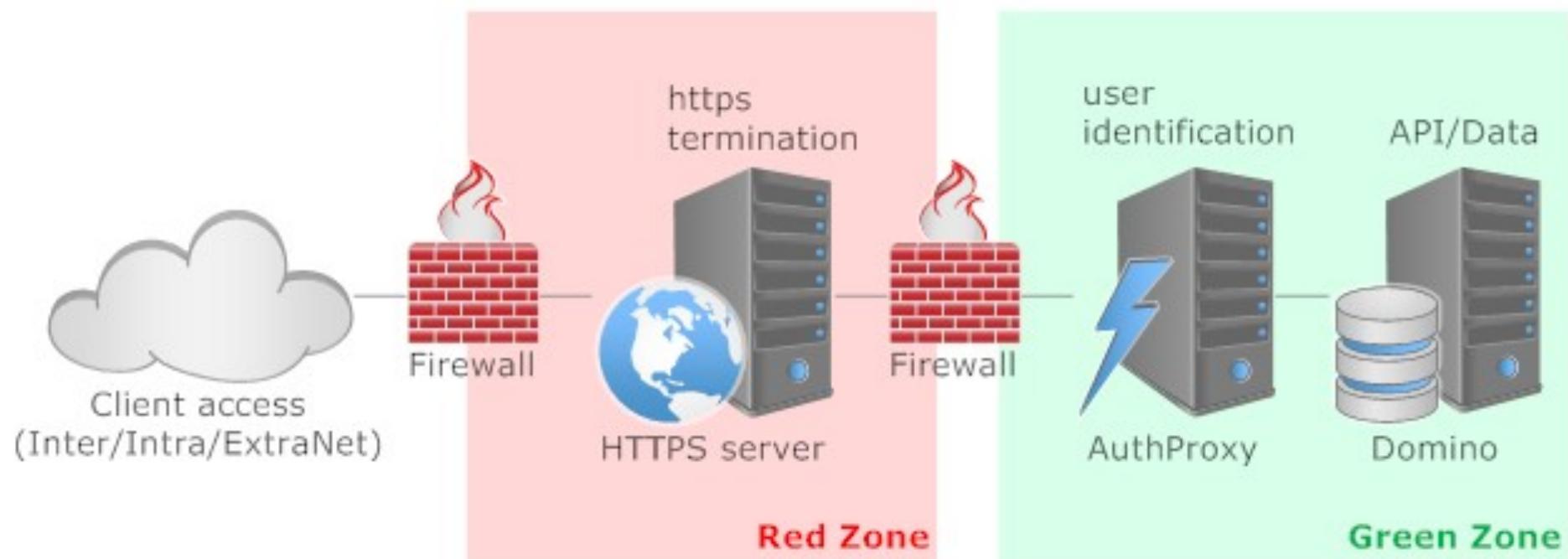
How

HOW

Deployment Option 1



Deployment Option 2



Comparing deployment options

Single NSF

- Easy deployment
- Domino HTTP stack
- Domino authentication
- Single tier

Front-end / Back-end split

- Easy update of frontend
- http2 capable
- Any authentication
- Multi tier

There is no “one perfect solution”, only choices and consequences



Client - Server API

What I want to send/receive to/from the server:

```
{    "color" : "green",  
    "taste" : "sour",  
    "name"  : "Apple"  
}
```

JSON rulez supreme!
XML is out of fashion
x-www-form-urlencoded
Is kind of lame



Backend options

- Agents (if you are still on LotusScript)
- XAgents (hardcore SSJS)
- Just views (read only)
- XPages Rest controls
- Domino Access Services
- Custom servlets (Wink or otherwise)
- **SmartNSF** (highly recommended!!!)
- External Java8 (I like vert.x)

There is no “one perfect solution”, only choices and consequences

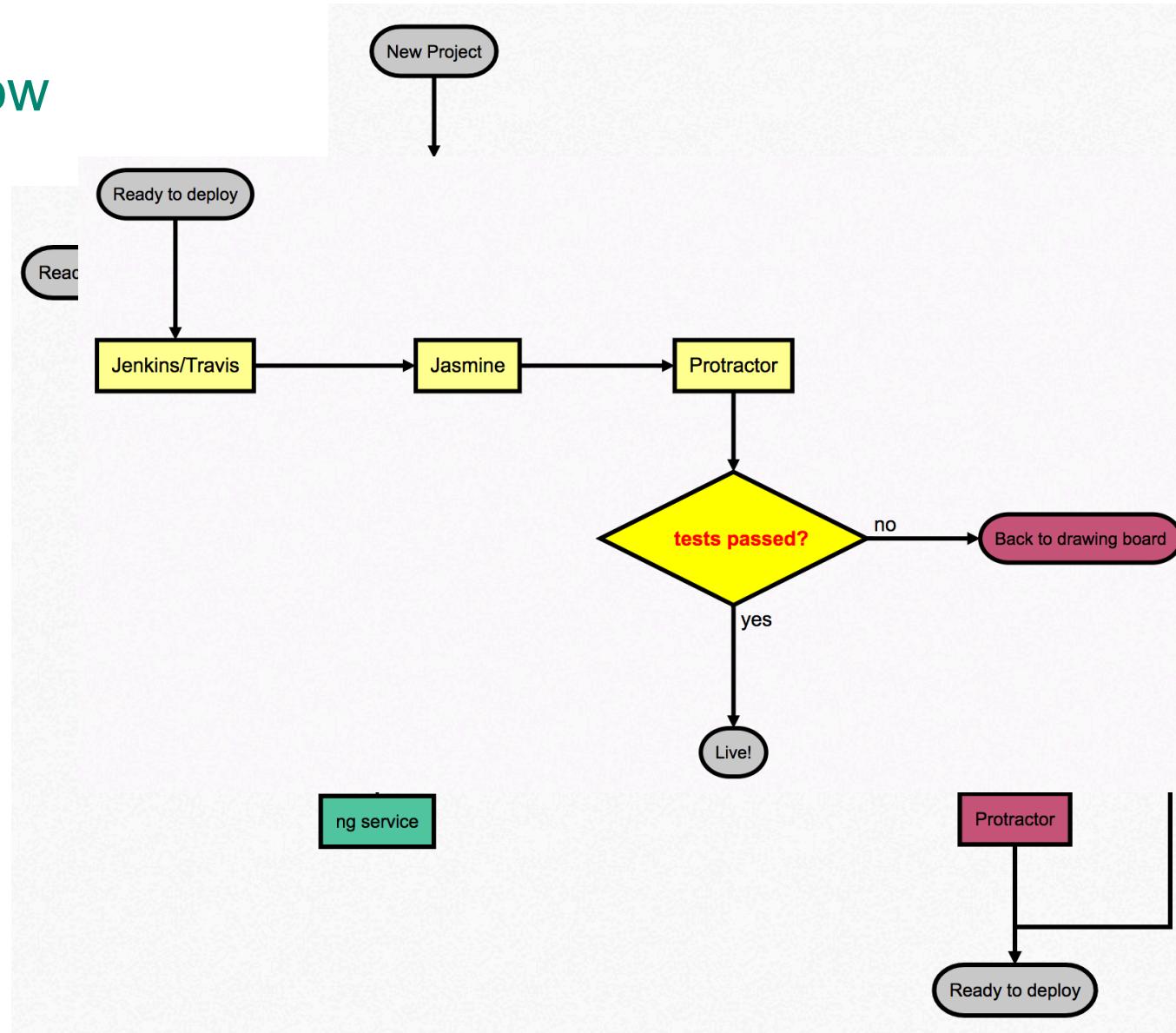
Pro tip:
Be lean on your API



Developer workflow

Easy as 1-2-3

1. Scaffold
2. Code
3. Deploy



Put Domino into CI?

Pro:

- Fresh start
- Known state

Con:

- Heavy operation
- NSF updates hard

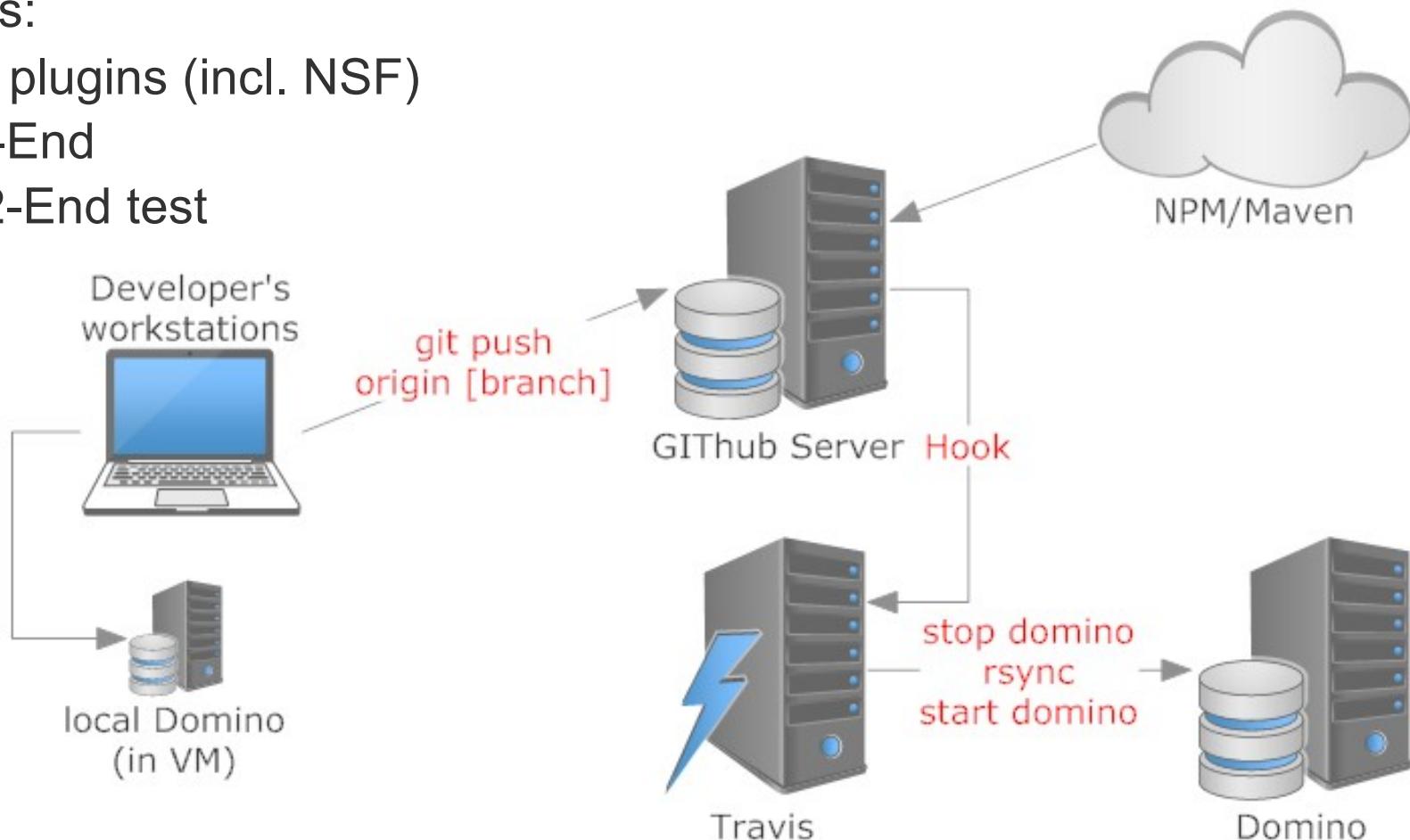
Key consideration:
Do you have date
sensitive operations?

- New documents arrived
- Documents changed
- Test design!



A potential setup

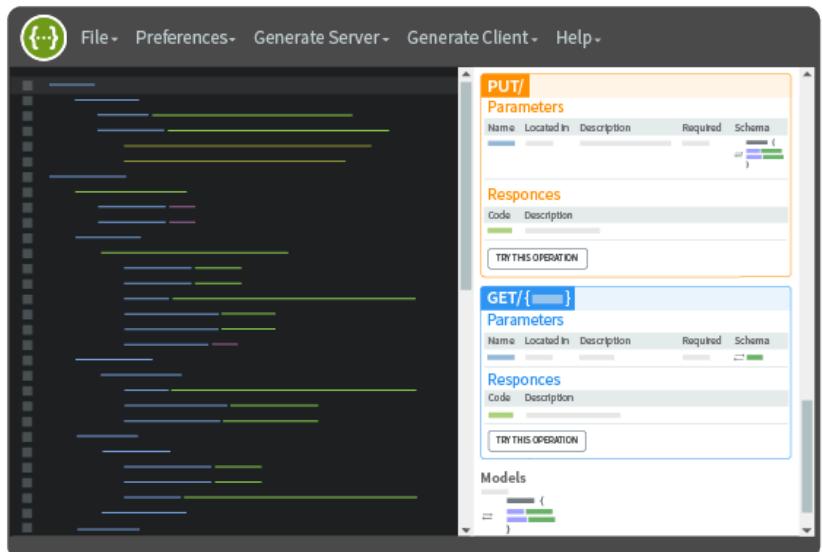
- 3 Projects:
 - OSGi plugins (incl. NSF)
 - Front-End
 - End-2-End test



Data model

Classic Domino

- Forms
- Views



API Driven Web applications:

- Swagger
- Data model
- Action model
- Code generation

HOW

Our Swagger model

33 2/22/17

```
1  swagger: '2.0'
2  info:
3    version: '1.0.0'
4    title: Domino foodie list
5    description: A sample API for Connect 2017
6    termsOfService: https://notessensei.com/
7    contact:
8      name: The NotesSensei
9      email: stephan@wissel.net
10     url: https://notessensei.com/
11    license:
12      name: Apache-2.0
13      url: https://opensource.org/licenses/Apache-2.0
14  host: food.projectcastle.io
15  basePath: /api
16  schemes:
17    - http
18  consumes:
19    - application/json
20  produces:
21    - application/json
22  paths:
23    /food:
24      get:
25        description: Returns the list of all food items
26        operationId: findFood
27        produces:
28          - application/json
29        parameters:
30          - name: type
31            in: query
32            description: filter the food type
33            required: false
34            type: array
35            items:
36              type: string
37              collectionFormat: csv
38          - name: rows
39            in: query
40            description: maximum number of rows to return
41            required: false
42            type: integer
43            format: int32
44        responses:
45          '200':
46            description: food response
47            schema:
48              type: array
49              items:
50                $ref: '#/definitions/food'
51          default:
52            description: unexpected error
```

Domino foodie list

A sample API for Connect 2017

Version 1.0.0

Contact information

The NotesSensei

stephan@wissel.net

https://notessensei.com/

Terms of service

https://notessensei.com/

License

Apache-2.0

Paths

/food

GET /food

Description

Returns the list of all food items

Parameters

Name	Located in	Description	Required	Schema
type	query	filter the food type	No	string
rows	query	maximum number of rows to return	No	integer (int32)

Responses

Code	Description	Schema
200	food response	food { }

HOW

Swagger Model

```
definitions:  
  food:  
    type: object  
    required:  
      - name  
    properties:  
      id:  
        type: string  
      name:  
        type: string  
      taste:  
        type: string  
      color:  
        type: string  
      shape:  
        type: string  
      type:  
        type: string
```

Mock server to keep the front-end guys happy

```
app.js
1 'use strict';
2 var express = require('express');
3 var middleware = require('swagger-express-middleware');
4 var path = require('path');
5 var app = express();
6 var FileDataStore = middleware.FileDataStore;
7 var store = new FileDataStore();
8 middleware(path.join(__dirname, '../frontend/foodstore.yaml'),
9   app, function(err, middleware) {
10   app.use(
11     middleware.metadata(),
12     middleware.CORS(),
13     middleware.files(),
14     middleware.parseRequest(),
15     middleware.validateRequest(),
16     middleware.mock(store)
17   );
18   app.listen(8000, function() {
19     console.log('The Swagger mock server is now running at http://localhost:8000');
20   });
21 });

HOW
```

Planning the Angular FrontEnd

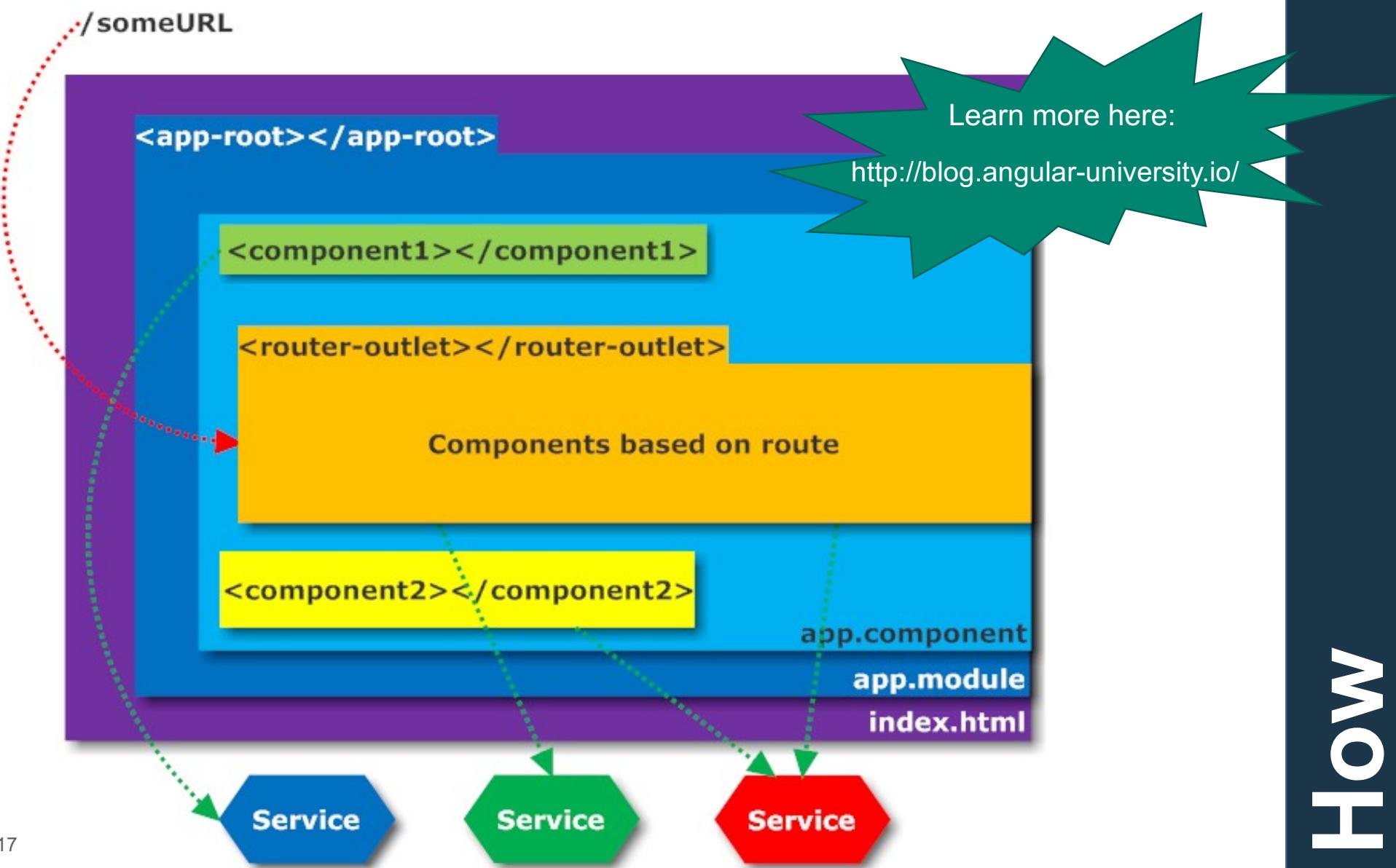
- Quick planning, refactor later
- Plan routes
 - One per view
 - Subroutes for documents
- Plan “Views”
 - One per view
 - One per document

That's just to get you started. Once you are comfortable, refactor!



How

Anatomy of an Angular App



Testing – double the work, a magnitude less headache

- OOTB: errors are reported on command line only

- Better reporting tool (1):

```
npm install --save-dev karma-mocha-reporter  
add require('karma-mocha-reporter') to plugins in  
karma.conf.js and reporters: ['mocha']
```

- Better reporting tool (2):

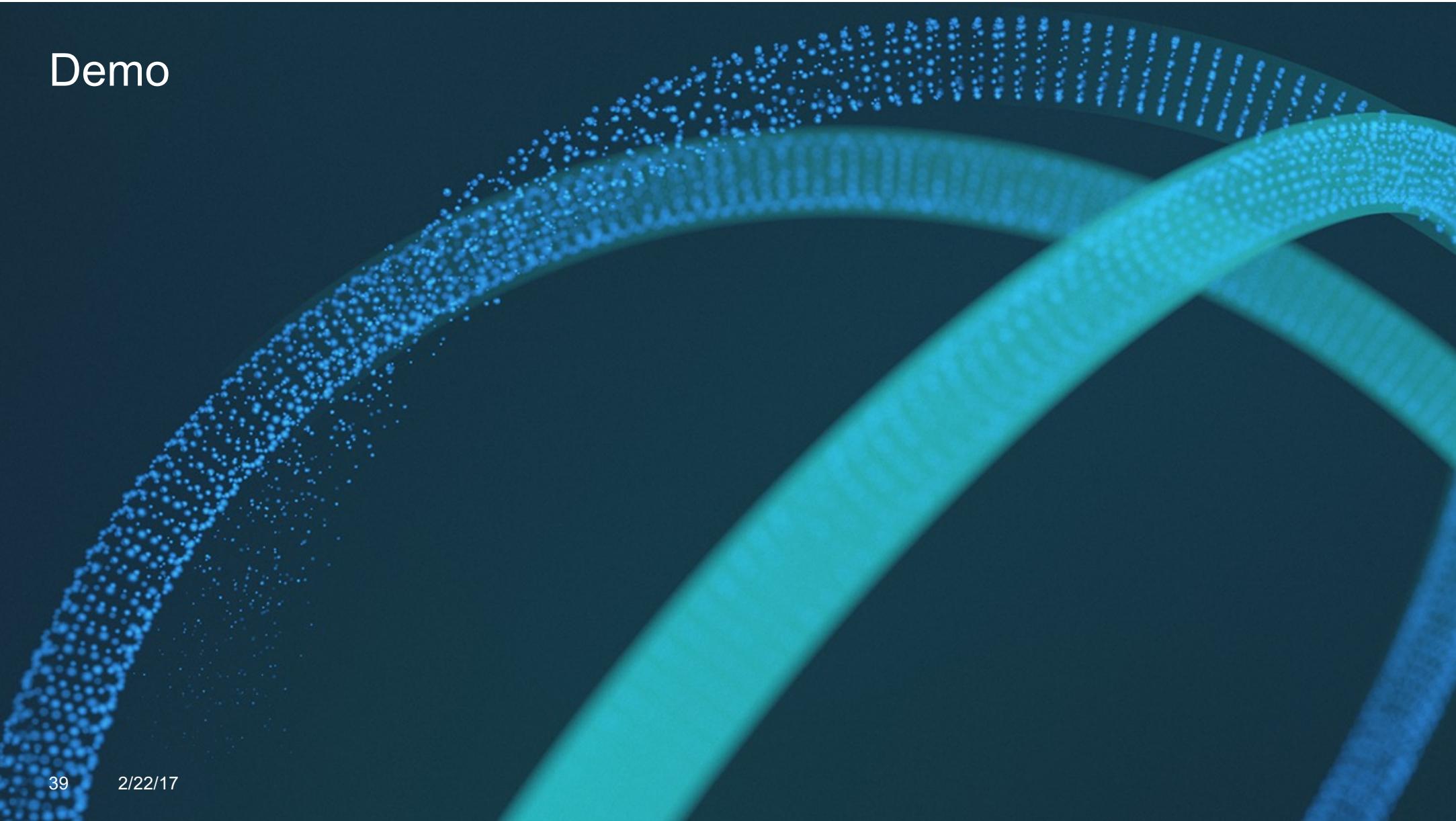
```
npm install --save-dev karma-htmlfile-reporter  
add require('karma-htmlfile-reporter') to plugins  
in karma.conf.js and reporters: ['mocha', 'html']
```

For CI consider JUNIT
report format. Integrates
best with most CI reports



HOW

Demo



All this command line stuff vs. Designer



Code Whisperers don't
need a fancy GUI. They
talk to the command line



Source: https://en.wikipedia.org/wiki/Control_of_fire_by_early_humans

Testing – some insights

- You will edit your spec files frequently
- Each new component needs to be added to spec
- Mocks are a code smell (could be fragrant or stinky)
– use them wisely for services!
- Consider <https://wallabyjs.com/> to continuously run your tests while typing*
- Providers can be tricky (consider Mocks for some)
 - {provide: Router, useClass: RouterModule}
 - {provide: Http, useClass: HttpModule}
- Takes quite some getting used to

* bad when on battery

In Summary

- Contemporary web development requires node.js as tooling
- Version control is your friend
- Make peace with the command line: npm, ng, git, travis ...
- Http knowledge required. Use POSTMAN for convenience
- Separate front-end and back-end with a strong API
- Continuous integration & Test Driven development rule

There is a lot to
learn in new world
of web development



The background of the slide features a dark teal or black gradient. Overlaid on this are two glowing blue particle streams. One stream forms a diagonal band from the bottom left to the top right. The other stream is more curved and horizontal, extending from the middle left towards the bottom right. Both streams consist of numerous small, glowing blue dots.

Questions?

<https://github.com/Stwissel/connect2017-dev1545>

Thank you!

Notices and disclaimers

Copyright © 2017 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and disclaimers continued

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services ®, Global Technology Services ®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli®, Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Further readings

Would you like to know more?*

The full deck & all the files

The presentation & sample files:

<https://github.com/Stwissel/connect2017-dev1545>

The Swagger Codegen update:

<https://github.com/Stwissel/swagger-codegen>

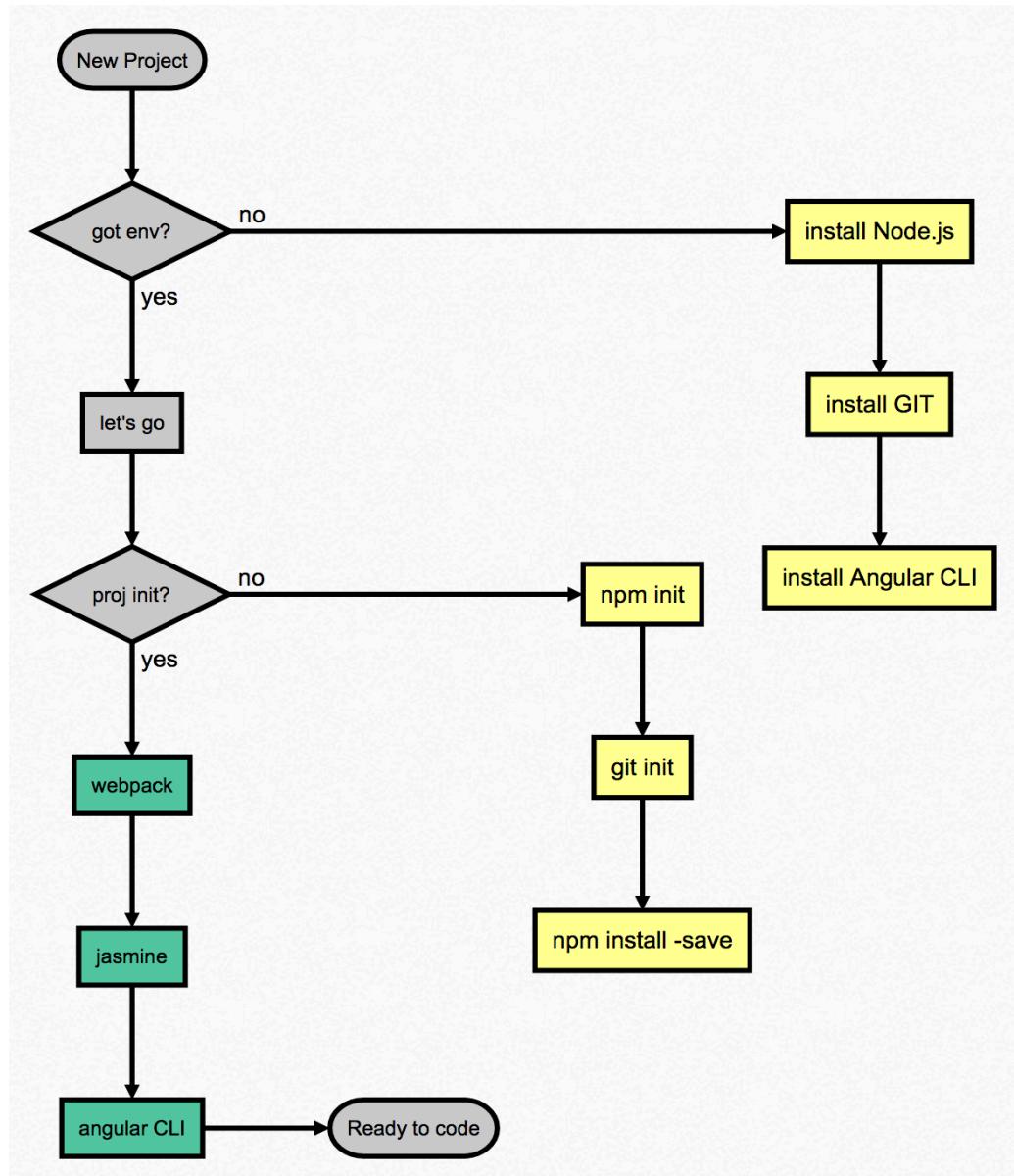
Formly for Angular:

<https://github.com/formly-js/ng-formly>

Tool locations

- Node.js: <https://nodejs.org/>
- NPM: <https://www.npmjs.com/>
- Jasmine: <https://jasmine.github.io/>
- Protractor: <https://www.protractortest.org/>
- AngularJS: <https://angular.io/>
- Angular CLI: <https://cli.angular.io/>
- TypeScript: <https://www.typescriptlang.org/>
- Webpack: <https://webpack.js.org/>
- Microsoft Code: <https://code.visualstudio.com/>
- Swagger: <http://swagger.io/>
- Wallaby: <https://wallabyjs.com/>
- Postman: <https://www.getpostman.com/>
- GIT: <https://git-scm.com/>
 - Tower: <https://www.git-tower.com/>
 - Sourcetree: <https://www.sourcetreeapp.com/>
- Mustache: <https://mustache.github.io/>

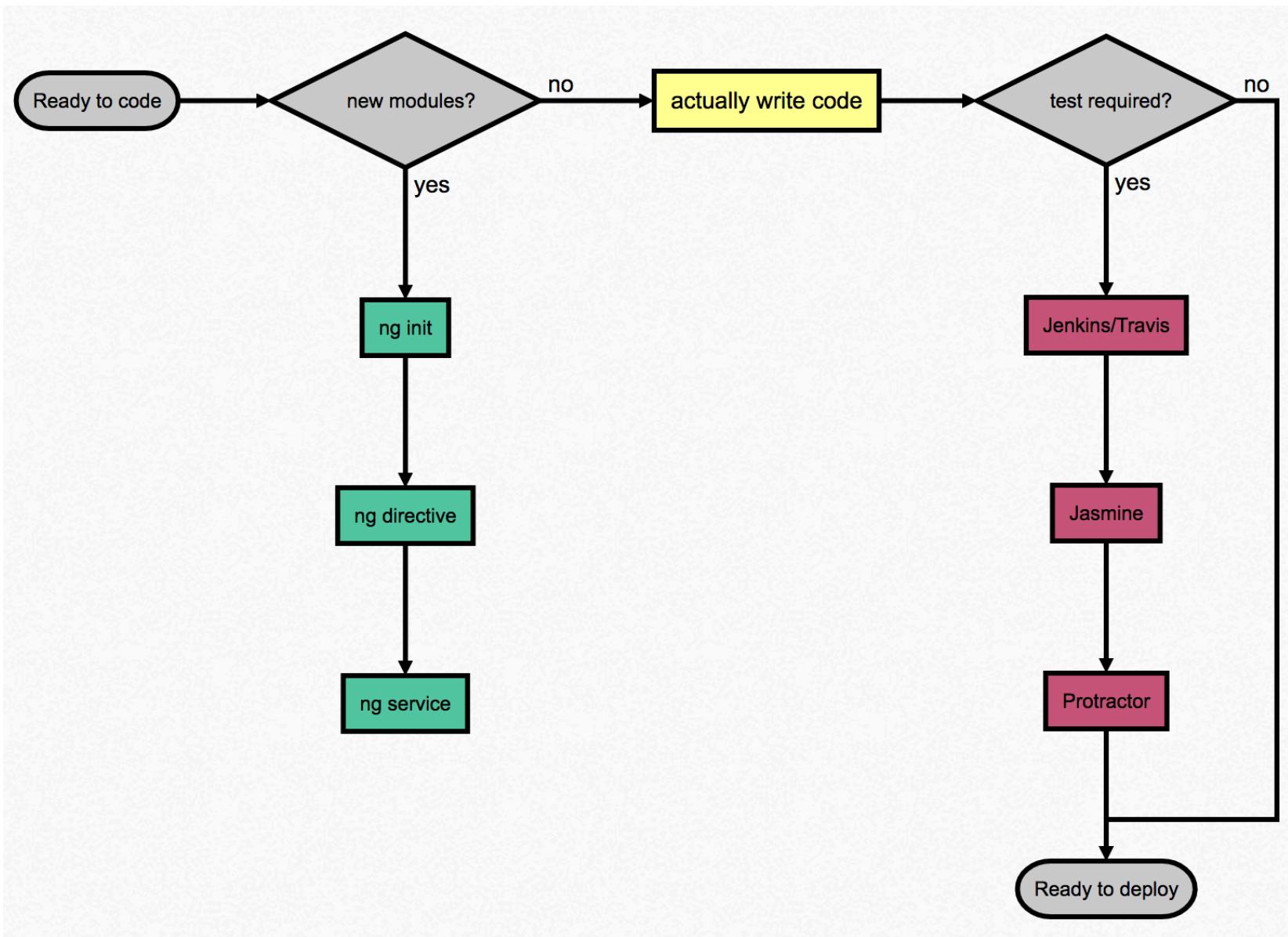
Developer workflow (1/3)



Developer workflow (2/3)

2/22/17

50



Developer workflow (3/3)

