

AngularJS

Angular

Brown Bag Seminar - January 2015

why Angular?

- HTML is static
- intermingled Javascript makes it dynamic
- add AJAX and it's interactive

frameworks abstracting this away are complex

- decouple DOM from the application logic
- retain HTML as layout - declarative programming
- provide structure for imperative programming

for example...

[quick demo]

it's MVC

model - javascript

view - html

controller - javascript

... with business logic on the server via
RESTFul APIs

how does it work?

Angular consumes the DOM and ties it to the Javascript model and controller

why so interesting?

single page

MVC

double binding

RESTFul *JSON of course*

pure Javascript

what else?

- filters
- modules
- routes
- dependency injection
- factories
- ... and also

tooling out of the box!

dependency management

test-driven development

continuous integration

more app

Angular and RESTEasy fit nicely

what can we use?

many controllers to a view

can we try this with JSF?

- one controller per tab, one per include, ...
- recognize Javascript code as a controller

some deficiencies

based on Javascript

- no classes
- not typesafe

no architectural enforcement

what's next

AngularJS 2.0

ECMAScript 6

today's Javascript is ECMAScript 5.1

more information

Dan Wahlin's talk

<https://www.youtube.com/watch?v=tnXO-i7944M>

video tutorial

<https://egghead.io/technologies/angularjs?order=ASC>

interactive tutorial

http://www.w3schools.com/angular/angular_intro.asp

versioned tutorial

<https://docs.angularjs.org/tutorial>