Grails REST + Spine.js

Decoupling Grails and Front End Apps

Who are we?

Aaron Eischeid

- Lead Developer at GlobalVetLINK
- @aeischeid

Craig Atkinson

- Sr. Consultant at Object Partners, Inc.
- craig.atkinson@objectpartners.com

Web trends

The rise of Web Services

- you're not cool if you don't have an API

Pages are becoming Applications

- HTML5

Javascript is 'fast' now

- opening the door to using it like a 'real' programming language

Benefits of using web services

Adding what others have done well into your application via web services

- Ex. Google Maps, Amazon S3

Data portability between web applications

Web services technologies

Several technologies for web services

- REST, SOAP, RPC, etc.

Many popular web services use REST

- Google, Twitter, GitHub

What is a REST API?

Interact with application via HTTP requests

- GET, POST, etc.

Endpoint URLs for specific actions

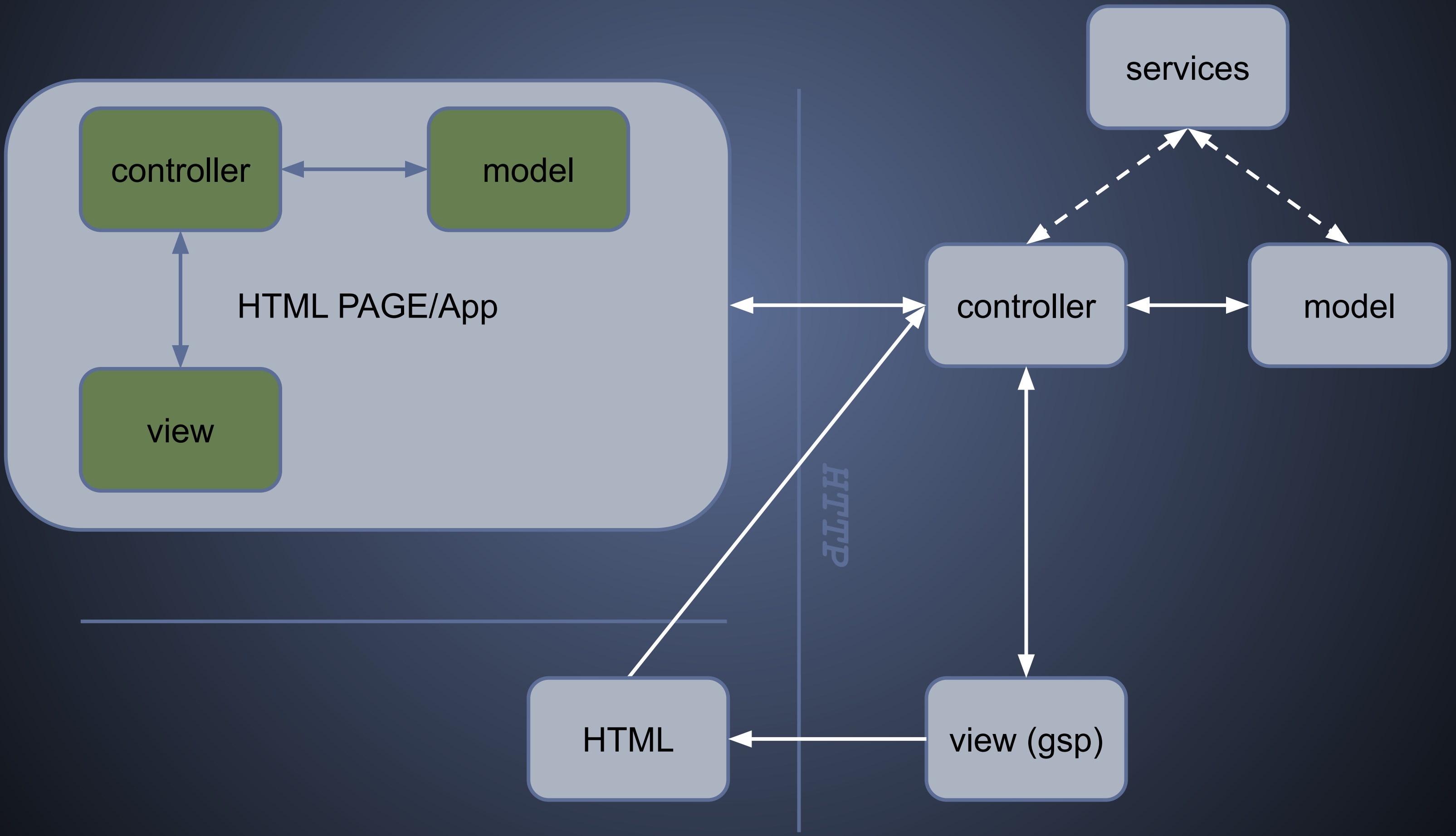
- GET statuses/retweeted_by_me

Client/server web app with REST

Combine front-end JavaScript app with back-end web application

Communicate via JSON REST API

What is client side MVC?



Client-side state benefits

Web app "feels" more like native app

Remove standard wait for request-response cycle

- Especially long in mobile

The door is open for 'offlining' your web app

Eating own dogfood

JavaScript front-end apps use same REST API as third parties

API in core of application, not afterthought

JavaScript front-end app

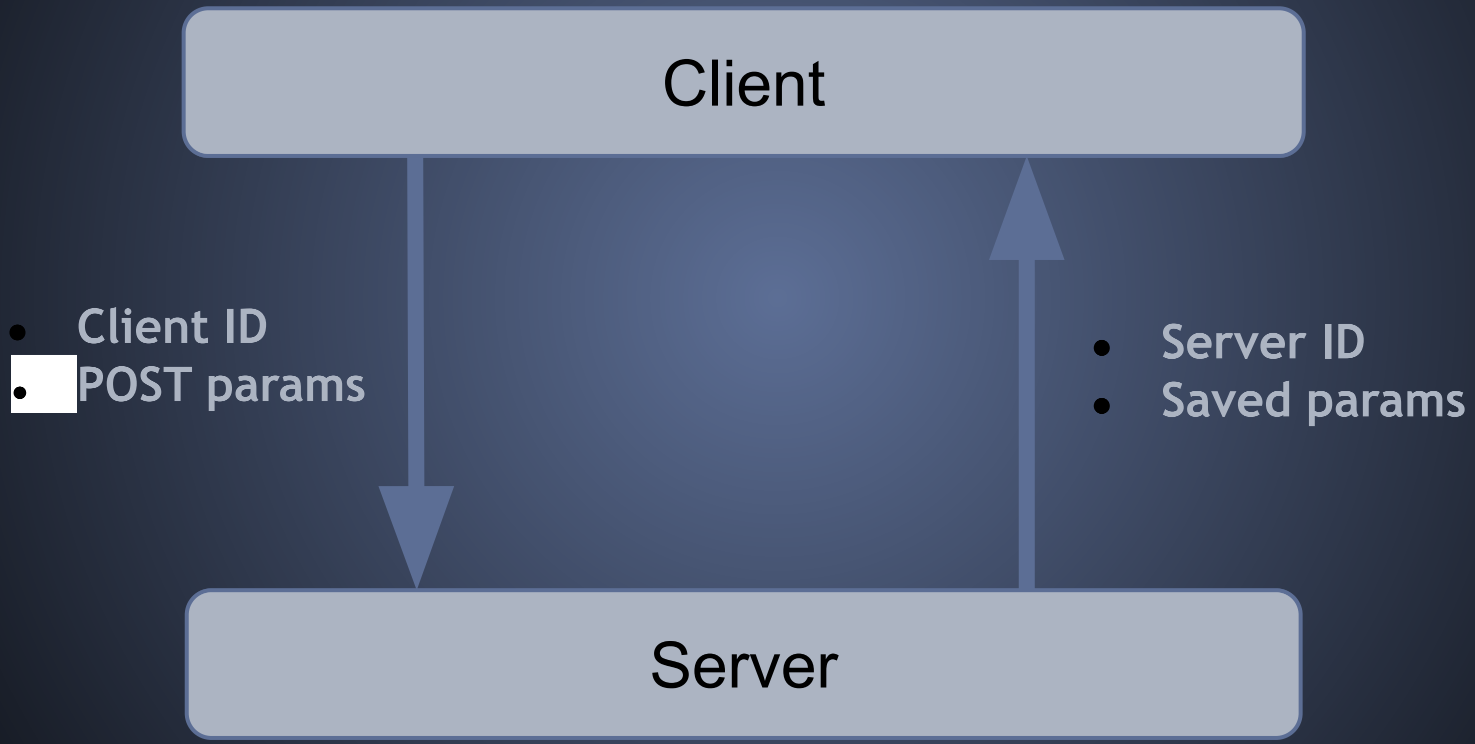
Non-blocking interface (async UI)

Models live on client and server

- Persistence via Ajax or LocalStorage

Elegant client-side event binding and management

New object round trip



Client

- Client ID
- POST params

- Server ID
- Saved params

Server

What is Spine?

"...the first JS MVC-type library that hasn't made me want to slam my hand into a drawer"

-*Graham Ballantyne on the spine group mailing list*

- Convention over configuration
- Uses a developer friendly language (or not...)

similar to...

GRAILS

RAILS

http://www.spinejs.com/

Spine vs. ... vs. ???

Frontend frameworks popping up all over.

Great comparison project using todo apps

Decoupling means less lock in!

Spine is currently one of the best in our opinion, but if tomorrow some new awesomeness arises we can move to that so long as it adapts to our API

Spine preprocessors

CoffeeScript -> JavaScript

Stylus -> CSS

Eco Templates...

CoffeeScript

JavaScript's better dressed more friendly alter-ego?

Easy to learn - http://coffeescript.org/

Play - coffeeConsole Chrome extention

Stylus

Similar to LESS or ...

Semicolons can still be friends ;)

- or not...

Variables, inheritance, oh my!

- easy to learn
- http://learnboost.github.com/stylus/

Eco Templates

The views!

Use Coffescript

Syntax similar to ERB

https://github.com/sstephenson/eco

JavaScript Testing

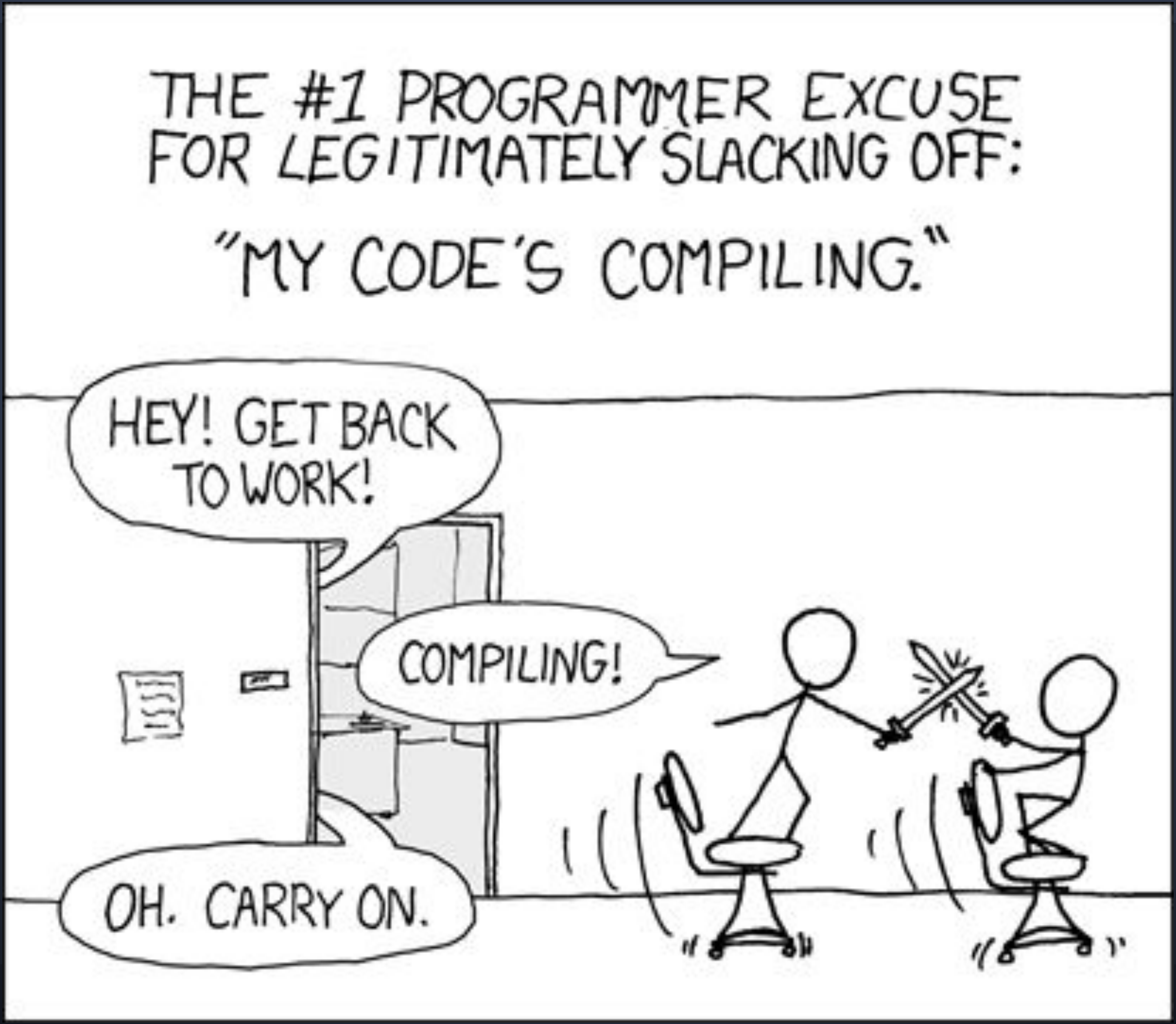
Mucho rapido!!!

- so fast to run you might actually write them?!

Jasmine

- http://pivotal.github.com/jasmine/
- http://evanhahn.com/?p=181

Compiling...


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.

Hem!

A node.js package

Makes developing with preprocessors painless

Optimizes production CSS/JS

Can even write tests in Coffescript!

Using hem

Running

hem server -> grails run-app

localhost:9294 - > localhost:8080

Testing

hem server -> grails test-app

localhost:9294/test ->

file://..
..GrailsProject/target/test-reports/html/index.html

Deploy

hem build -> grails war

or not...

At the end of the day Spine is just javascript and css, so you can chose to leave behind or replace some or all of the pieces.

For example:

Use whatever frontend templating solution you want.

- including .gsp's! (theoretically...)

Grails REST API

Server-side persistence and processing for JavaScript front-end app

Basic CRUD functions + custom API actions

Scaffolding

Grails generates dynamic, scaffolded code and views based on domain classes

Scaffolded controller useful place for domain class CRUD actions

Parent controller

Common parent controller provides basic CRUD operations

RESTful controllers extend and/or customize default actions

DRY REST API

Generic controller code for CRUD API for domain classes

Add custom, specialized API actions to expand on scaffolded CRUD actions

Map HTTP method -> Grails action

GET -> list(), show(id)

POST -> save()

PUT -> update(id)

DELETE -> delete(id)

Render JSON from Grails

Render JSON instead of HTML model/view

```
import grails.converters.JSON

class MyController {
  void myAction() {
    render myObject as JSON
  }
}
```

Customize JSON

Restrict which fields are returned in JSON (or create new fields)

```
JSON.registerObjectMarshaller(MyObject) {
  def jsonMap = [:]
  jsonMap.id = it.id
  jsonMap.firstName = it.first
  jsonMap.fullName = it.first + it.last
  return jsonMap
}
```

Render the right format

Render different formats from same controller action, based on client's format

```
void myAction() {
  def myObject = myService.findObject()
  withFormat {
    html  {
      render(view: "myView", model: [instance: myObject])
    }
    json {
      render myObject as JSON
    }
  }
}
```

API safety net

- Functionally test REST API with Grails Functional Test Plugin
  - Tests quick to write and fast to run

JSON in functional tests

```
String jsonString = new JSON(
   id: 1,
   firstName: "John",
   lastName: "Smith").toString()

JSONElement getParsedJsonElement() {
   JSON.parse(response.contentAsString)
}
```

Decoupling deployment option

Grails backend deployed in application server

- Tomcat, Jetty, etc.

Spine frontend deployed in webserver

- Apache, Nginx

Frontend changes deployed without requiring app server restart

Frontend apps and GSPs coexist

Use frontend apps where they make sense, GSPs otherwise

Can add frontend apps into existing Grails apps

Conclusion

Integrating apps via API increasingly popular

Decoupling has many benefits

- app performance (perception) and potential features
- separate development and deployment
- scalability

Grails & Spine.js: simple yet powerful app frameworks

Resources

Books

jQuery Developers' Guide to Moving State to the Client
JavaScript Web Applications
O'REILLY
Alex MacCaw

Web Services for the Real World
RESTful Web Services
O'REILLY
Leonard Richardson & Sam Ruby

Articles

http://alexmaccaw.com/posts/async_ui

http://blog.alexmaccaw.com/rails-is-just-and-api-and-that-s-ok

http://www.aaideas.com/2012/04/19/grails-and-spine-js-decoupled-development-setup/

http://www.aaideas.com/2011/11/22/customized-grails-controller-for-rest/

What is Disc Golf?

fun! :)

Demo App

Source:

https://github.com/aeischeid/DgScorecard

Deployed:

DgScorecard

API:

gDoc spreadsheet