```
//Feed your brain
gr8.technologies.each {
    yourBrain << it
}
```

Fork me on GitHub

Grails Worst Practices

Burt Beckwith

All of the topics discussed here are important, and *100%* serious

Important Concepts:

Easier is better

Important Concepts:

Finishing tasks faster is better

Important Concepts:

Use the first solution that you think of

Important Concepts:

Minimize the time you spend at work

Important Concepts:

Get others to do your work for you

Important Concepts:

Lazy is good

Important Concepts:

If there's a problem, fix it if and when you have to

Important Concepts:

Cover your tracks

Important Concepts:

Be more like your dog

http://cuteoverload.files.wordpress.com/2014/03/cute-smiling-animals-2.jpg

It's fine to disappear for 6 months, little will have changed

GET STARTED

- Start with Grails!
- Installation
- Quick Start
- IDE Setup
- Tutorials
- Screencasts

REFERENCE

- Documentation
- FAQs
- Roadmap

Last updated by graemerocher 4 months ago

Grails 2.2.5 Release Notes

Grails is a dynamic web application framework built on Java and Groovy, leveraging best of breed APIs including Spring, Hibernate and SiteMesh. Grails brings to Java and Groovy developers the joys of convention-based rapid development while allowing them to leverage their existing knowledge and capitalize on the proven and performant APIs Java developers have been using for years.

Release Information

- JIRA Changelog
- Download
- User Guide
- What's new in Grails 2.2 Guide

GET STARTED

- Start with Grails!
- Installation
- Quick Start
- IDE Setup
- Tutorials
- Screencasts

REFERENCE

- Documentation
- FAQs
- Roadmap

Last updated by jeff.brown 10 months ago

Grails 2.3.0 Release Notes

Grails is a dynamic web application framework built on Java and Groovy, leveraging best of breed APIs including Spring, Hibernate and SiteMesh. Grails brings to Java and Groovy developers the joys of convention-based rapid development while allowing them to leverage their existing knowledge and capitalize on the proven and performant APIs Java developers have been using for years.

Release Information

- JIRA Changelog
- Download
- User Guide
- What's new in Grails 2.3 Guide

GET STARTED

- Start with Grails!
- Installation
- Quick Start
- IDE Setup
- Tutorials
- Screencasts

REFERENCE

- Documentation
- FAQs

Last updated by graemerocher 2 months ago

2.4.0 Release Notes

Grails is a dynamic web application framework built on Java and Groovy, leveraging best of breed APIs including Spring, Hibernate and SiteMesh. Grails brings to Java and Groovy developers the joys of convention-based rapid development while allowing them to leverage their existing knowledge and capitalize on the proven and performant APIs Java developers have been using for years.

Release Information

- JIRA Changelog
- Download
- User Guide
- What's new in Grails 2.4 Guide

GET STARTED

- Start with Grails!
- Installation
- Quick Start
- IDE Setup
- Tutorials
- Screencasts

REFERENCE

- Documentation
- FAQs
- Roadmap

Last updated by graemerocher 1 month ago

2.4.2 Release Notes

Grails is a dynamic web application framework built on Java and Groovy, leveraging best of breed APIs including Spring, Hibernate and SiteMesh. Grails brings to Java and Groovy developers the joys of convention-based rapid development while allowing them to leverage their existing knowledge and capitalize on the proven and performant APIs Java developers have been using for years.

Update Notes

Recommended plugin versions

If you are upgrading from Grails 2.4.2 and you use the hibernate4 plugin, you will need to update the version in BuildConfig:

```
runtime ':hibernate4:4.3.5.4' // or ':hibernate:3.6.10.16'
```

recommended tomcat, asset-pipeline, cache and scaffolding plugin versions for Grails 2.4.x :

```
build ':tomcat:7.0.54'
compile ':cache:1.1.7'
```

Grails Can Be Frustrating

http://www.flickr.com/photos/alexchoi/351267249/

Try to think like a PHP developer

Facebook uses it, and they have over 1,000,000,000 users

Try to think like a PHP developer

Sure, we can use O-O principles, but we aren't required to, so why bother?

Controllers

One controller per domain class

It's how scaffolding is generated,
so it must be the best way

Corollary:

Always trust generated code

Put most of your code in controllers

They're your interface between
the client and the application

Do a lot of work in GSPs

If Grails doesn't want us to have Groovy code in scriptlets (and to call GORM, etc.) then why is it allowed?

Do a lot of work in GSPs

Eventually you'll hit the maximum GSP size limit, but that's a problem for later

Do a lot of work in GSPs

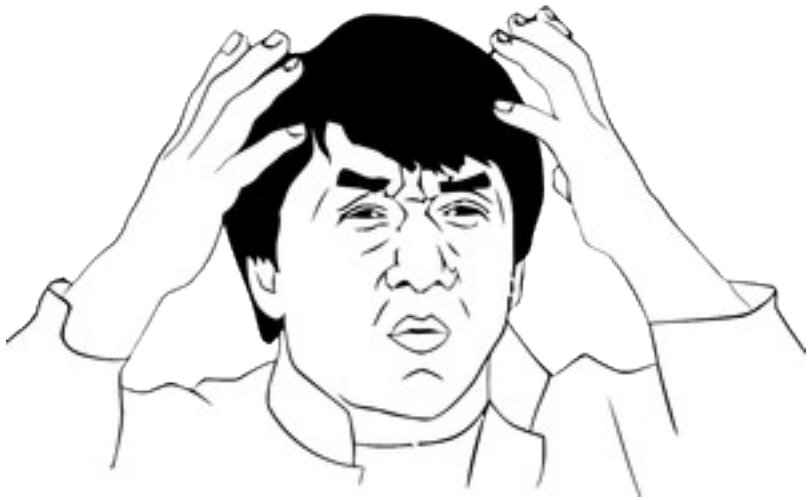
Don't worry that it's difficult to test code in GSPs

Do a lot of work in GSPs

"@BurtBeckwith What is the advantage of using TagLibs over writing code on the view?"

Do a lot of work in GSPs

"@BurtBeckwith What is the advantage of using TagLibs over writing code on the view?"



Transactions are important

Luckily for us, there's a cool new `@grails.transaction.Transactional` annotation that can be used in controllers. How cool is that?!?

Transactions are important

Unfortunately, **`@Transactional`** cannot be used in GSPs

Transactions are important

Someone should create a JIRA enhancement request for that.

Transactions are important

Someone should create a JIRA enhancement request for that.

Until then, use `withTransaction`

Don't waste time with services

If all of your business logic is in controllers, who needs services?

Don't waste time with services

Don't bother creating small, focused services

Don't waste time with services

And it would be silly to reuse or share them, e.g. in a plugin

Don't waste time with services

And it would be silly to reuse or share them, e.g. in a plugin

Don't waste time with services

And it would be silly to reuse or share them, e.g. in a plugin

It was fun to write once, so it'll be fun to write again and again

Idiomatic Groovy

Always use optional types

Use **`def`** everywhere. It's so easy to type.

Always use optional types

If the Groovy runtime can figure out what's going on, so can your coworkers.

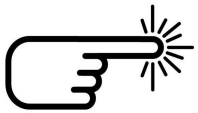
Prefer closures to methods

They will impress your friends
who don't know Groovy

Write the grooviest code possible

Don't bother rewriting performance-critical code in Java or in Groovy with **`@CompileStatic`**

Don't follow these guidelines:



Groovy style and language feature guidelines for Java developers

Code Reuse

Copy and paste

Copy and paste. A lot.

Copy and paste. A lot.

If it works, keep using it

Copy and paste. A lot.

When you copy and paste, you don't have to deal with taglibs, templates, helper classes, etc.

Copy and paste. A lot.

If it got a lot of votes on Stack Overflow, it belongs in your code

Copy and paste. A lot.

If buggy code was copied and pasted in several places, be sure to try to find and fix all of them

Testing

Testing is not worth it

I don't get paid more if I write more code, do you?

Testing is not worth it

You spend all that time writing tests, and then something changes and they fail

Testing is not worth it

If you don't have tests, you can't get yelled at for breaking the build

Delay testing where possible

If you must write tests, wait until the end of the project. There's always extra time for tests and documentation

Always write unit tests

If you must write tests, always use unit tests because they run the fastest.

Always write unit tests – especially for persistence

It is not at all important to test persistence with a real database

Always write unit tests – especially for persistence

If that were true, then why does Grails let you unit test domain classes?

Don't bother with CI servers

All they do is complain about failed builds, usually because of failed tests

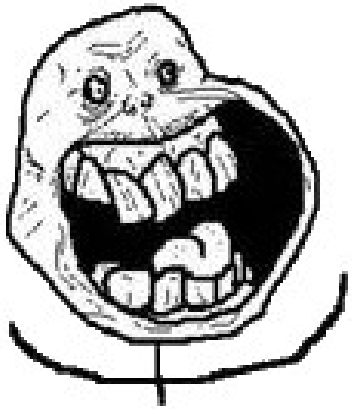
Security

Security is difficult

Don't bother with Spring Security or Shiro, just implement it yourself. It's fun and easy!

Security is difficult

Don't bother with Spring Security or Shiro, just implement it yourself. It's fun and easy!



Security is difficult

Always store passwords in the database without using complicated hashing algorithms

Security is difficult

It's important to be able to email users their passwords when they forget them

Security is difficult

And if you get hacked, it's not like they're going to steal your money, right?

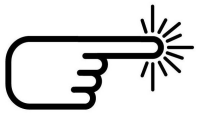
Store passwords in source control

Store passwords in source control

This is called "security by obscurity" and it always works

Store passwords in source control

I doubt that it's true that there are ~10,000 Amazon S3 production credentials in GitHub:



https://github.com/search?q=production+SECRET_ACCESS_KEY%3A+%22A&type=Code&ref=searchresults

Data Access

Always use `failOnError:true`

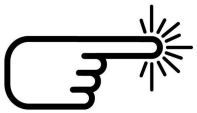
Don't bother checking `hasErrors()`. Let the error page handle expected user mistakes.

Always use `failOnError:true`

The users will probably figure it out. And besides, all they ever do is complain. They're so negative.

Always use `failOnError:true`

Exceptions aren't expensive,
especially not in Groovy



“The Exceptional Performance of Lil' Exception”

Always use `failOnError:true`

```
def thing = new Thing(params)
try {
   thing.save(failOnError:true)
   // handle success case
}
catch (e) {
   // handle error case
}
```

```
def thing = new Thing(params)
thing.save()
if (thing.hasErrors()) {
   // handle error case
}
else {
   // handle success case
}
```

Always use `failOnError:true`

```
def thing = new Thing(params)
try {
   thing.save(failOnError:true)
   // handle success case
}
catch (e) {
   // handle error case
}
```

```
def thing = new Thing(params)
thing.save()
if (thing.hasErrors()) {
   // handle success case
}
else {
   // handle error case
}
```

Obviously better

Database transactions are difficult

Database transactions are difficult

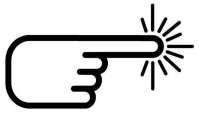
But you've probably heard that
they're important. I sure have.

Database transactions are difficult

Is there even any helpful information available?

Database transactions are difficult

Is there even any helpful information available?



http://2013.gr8conf.eu/Presentations/Grails-Transactions

GORM for REST

Speaking of convenience, have you seen GORM for REST? It's almost like giving your users direct access to your database:

```
import grails.rest.*

@Resource(uri='/books')
class Book {
   String title
}
```

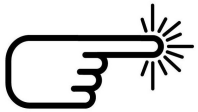
If you do use REST ...

Don't use

http://grails.org/plugin/
spring-security-rest

Always use collections for one-many and many-many relationships

Always use collections for one-many and many-many relationships



http://www.infoq.com/presentations/GORM-Performance

Cache as much as possible

Cache Everything

If a little caching is good, then a lot is great

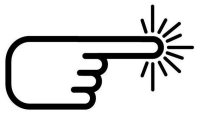
Cache Everything

But don't overthink it – just use

`grails.hibernate.cache.queries=true`

in Config.groovy

Cache Everything



Don't worry that
Hibernate query cache considered harmful?
appears to contradict this advice

Don't bother optimizing queries

It's better to make several queries that are easy to understand, and sort and filter the data in your code

```
def getAverageAgePerPlan() {

   def planAgeBarData = []

   Insurer insurer = ...

   insurer.plans.each { plan ->
      int ageTotal = 0
      for (Person person : plan.customers) {
         ageTotal += person.age
      }

      int avgAge = plan.customers.isEmpty()
        ? 0 : ageTotal / plan.customers.size()

      planAgeBarData << [plan.name, avgAge]
   }

   planAgeBarData
}
```

Don't share

Don't share

Don't write blog posts with information that others could use

Don't share

Don't create plugins – users will only complain about bugs and missing features

Don't share

Let them figure things out on their own, just like you had to.

Don't share

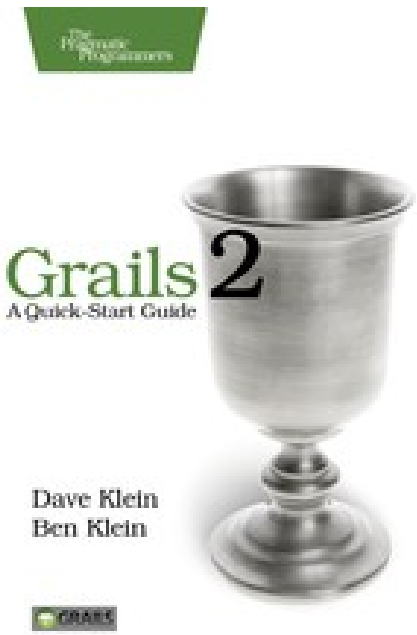
Don't report bugs (someone else will at some point)

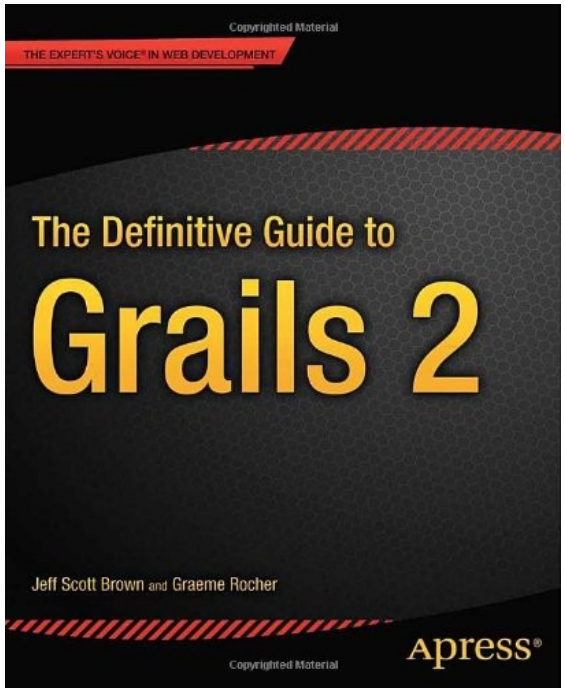
Don't share

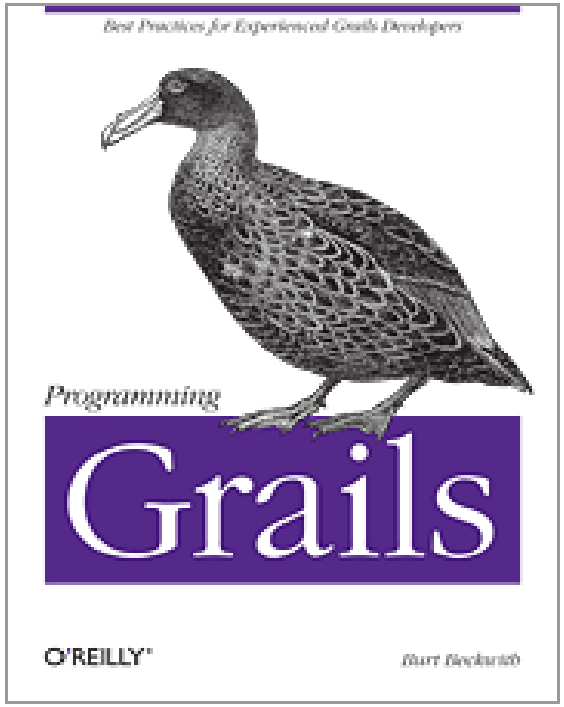
And if you do report a bug, don't bother trying to fix it with a patch or pull request

Don't bother reading books

Don't bother reading books







Don't bother with the mailing lists

Don't bother with the mailing lists

http://grails.org/Mailing%20lists

Don't bother with the mailing lists

http://grails.org/Mailing%20lists

http://grails.1312388.n4.nabble.com/

Don't bother with the mailing lists

http://grails.org/Mailing%20lists

http://grails.1312388.n4.nabble.com/

http://grails.markmail.org/search/?q=

Don't bother reading release notes and "What's New" pages

Don't bother reading release notes and "What's New" pages

http://grails.org/Release+Notes

What's new in Grails 2.3?

Don't Follow @grailsframework

Don't Follow @grailsframework

It's mostly a bunch of Graeme's Instagram photos of his lunch

Some miscellaneous items:

Some miscellaneous items:

Don't bother using packages, just put everything in the default package

Some miscellaneous items:

Don't worry about documenting your code

Some miscellaneous items:

Don't worry about writing self-documenting code

Some miscellaneous items:

Prefer jar files in the lib directory to dependencies in BuildConfig.groovy

Some miscellaneous items:

Prefer println to logging

Some miscellaneous items:

Store whatever you want in the HTTP session, especially domain class instances and query results. Memory is inexpensive.

And whatever you do ...

Don't buy this book:

O'REILLY®

Programming Grails

O'Reilly Media spreads the knowledge of innovators through its books, online services, magazines, research, and conferences. Get the information you need from the experts you trust; visit **oreilly.com** to purchase this book.

Get 40% off of the print book and 50% off of the ebook version of this book by entering discount code AUTHD.

Programming Grails
By Burt Beckwith
May 2013, ISBN 978-1-4493-2393-6
364 pages, $44.99

Spreading the knowledge of innovators | oreilly.com

¡Gracias!

http://cuteoverload.files.wordpress.com/2014/03/cute-smiling-animals-251.jpg