

# The Groovy Web:

*groovlets, microframeworks, and Grails*

# Who Am I?

Ken Kousen

President, Kousen IT, Inc.

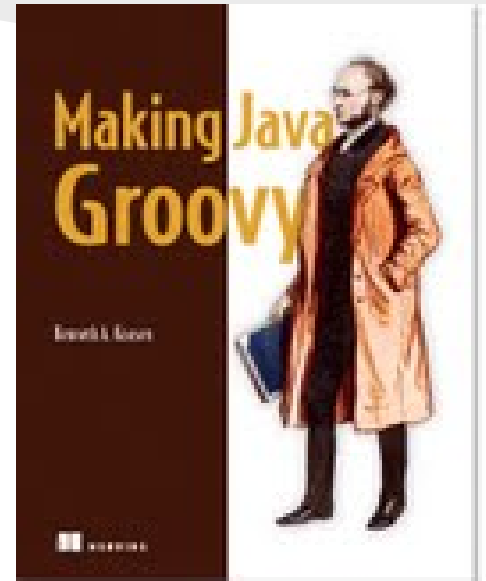
<http://www.kousenit.com>

[ken.kousen@kousenit.com](mailto:ken.kousen@kousenit.com)

[@kenkousen](#)

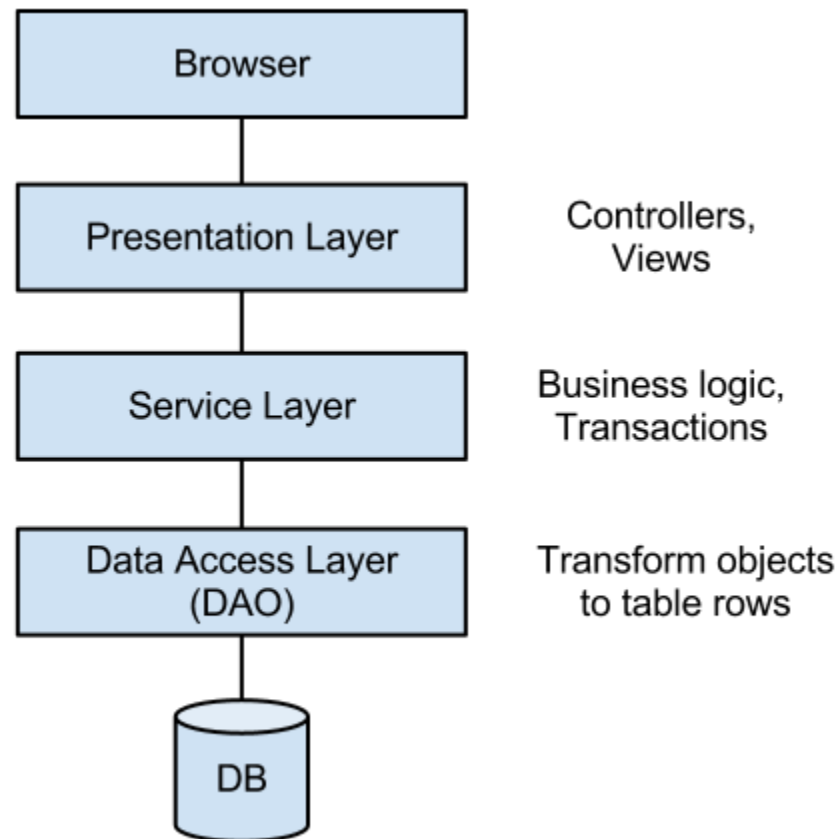
Making Java Groovy

<http://manning.com/kousen>



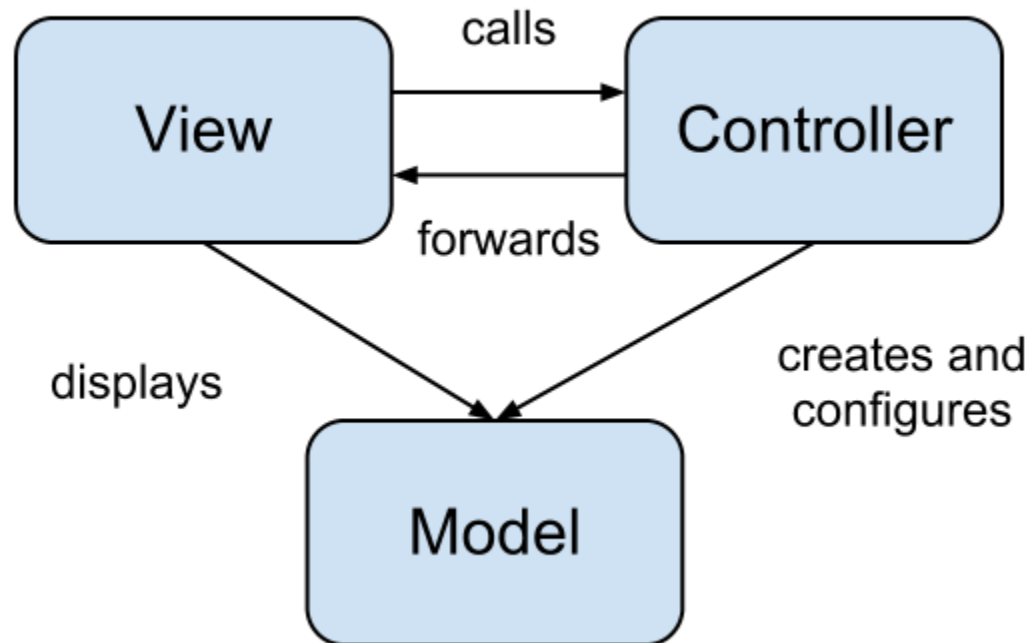
# Web Architecture

Every Java web app from 30,000 ft



# MVC

Oversimplification, but interesting



# Servlets

Java Servlets

Implement in Java or Groovy

# Servlets

Demo:

HelloServlet

HelloGroovyServlet

# ServletCategory

groovy.servlet.ServletCategory

## Groovy metaprogramming in action

```
static Object get(..., String key)
static void set(..., String key, Object value)
static void getAt(..., String key)
static void putAt(..., String key, Object value)
```

# ServletCategory

Demo:

HelloServletCategory



# Groovy Templates

Simple template engine

GString template engine

XML template engine

# groovlets

## Configure GroovyServlet

```
<servlet>
  <servlet-name>GroovyServlet</servlet-name>
  <servlet-class>
    groovy.servlet.GroovyServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>GroovyServlet</servlet-name>
  <url-pattern>*.groovy</url-pattern>
</servlet-mapping>
```

# groovlets

## Implicit variables

`request, response, session`

`application`

`params, header`

`html --> MarkupBuilder`

# groovlets

Convenient for simple applications

Easy for RESTful services

# groovlets

Demo:

HelloGroovlet

GroovyBaseball

SongService

# Ratpack

Microframework

Based on Sinatra (from ruby)

Build routes as URLs

Bundled with jetty

# Ratpack

Demo:

Basic routing

# Grails

Groovy's killer app  
(or Gradle, or Spock, or Griffon, ...)



Complete stack framework

Groovy DSLs +

Spring + Hibernate + SiteMesh

Tons of plugins



# Grails

## Architectural decisions

### *No DAO Layer*

DAO methods created by metaprogramming  
static methods on domain classes

```
Task.findByStartDateGreaterThan(...)
```

# Grails

Scaffolding

Controllers use DAO methods

# Grails

## Services

Transactional by default

Can use Spring `@Transactional`

Autowired by name into other artefacts

# Grails

Demo:

Quest for the Holy Grails

# Grails

Need to be careful

- Don't do too much in controllers

- Inject services

  - Let them do the work

- Don't bypass data access layer

  - (just because it's easy to do)

# Conclusions

Groovy simplifies Java

- Simpler syntax

- Builders and metaprogramming

groovlets useful for REST

Microframeworks show promise

Grails rocks, but be careful...

# Session Evaluations

Please complete your session evals :)