



Grails (グレイルズ)

See the light

アジャイル開発、業務に使えるWebアプリケーション開発、
ラピッドWebアプリケーション開発を簡単に。



Grailsとは？

- ◆ Ruby on Railsに触発された
Groovy,Javaによる、MVC Webアプリケーションフレームワーク
- **規約によるコーディング - Convention over Configuration**
- **Don't Repeat Yourself (重複を排除)**
- **Java資産の活用**



信頼できるオープンソース技術で構成

◆ 強固な地盤に構築されたフレームワーク

- **Spring**

強力なinversion of controlとMVC

- **Hibernate**

成熟し安定したオブジェクトリレーショナルマッピング

- **SiteMesh**

フレキシブルなレイアウトコントロールとページ・デコレーション

- **Groovy**

メタプログラミング、クロージャサポート、
構文的に表現力ゆたか、Javaの統合。



はじめてみよう！

環境の設定

◆Javaの設定

JDK1.4以上をインストール

環境変数設定

`JAVA_HOME=Javaのインストールパス`

◆Grailsの設定

最新版をGrailsサイトからダウンロード

`GRAILS_HOME=Grailsの解凍パス`



はじめてみよう！

Grailsは、コマンドを実行してアプリケーションの発行開発をします。

例) # **grails** **create-app** **book**

↑
Grails本体

↑
コマンド

↑
引数

Grailsのコマンド

create-app

create-domain-class

create-controller

create-sevice

create-job

create-plugin

run-app

war

generate-view

generate-controller

generate-all

generate-webtest

create-test-suite

create-tag-lib

install-plugin

他多数.....



初めてのアプリケーション

アプリケーションプロジェクトを作成

以下をコマンドで実行

```
# grails create-app book
```

実行すると、

フォルダと基本リソースが作成されます。



Grails

ディレクトリ構成

Grailsリソース関連

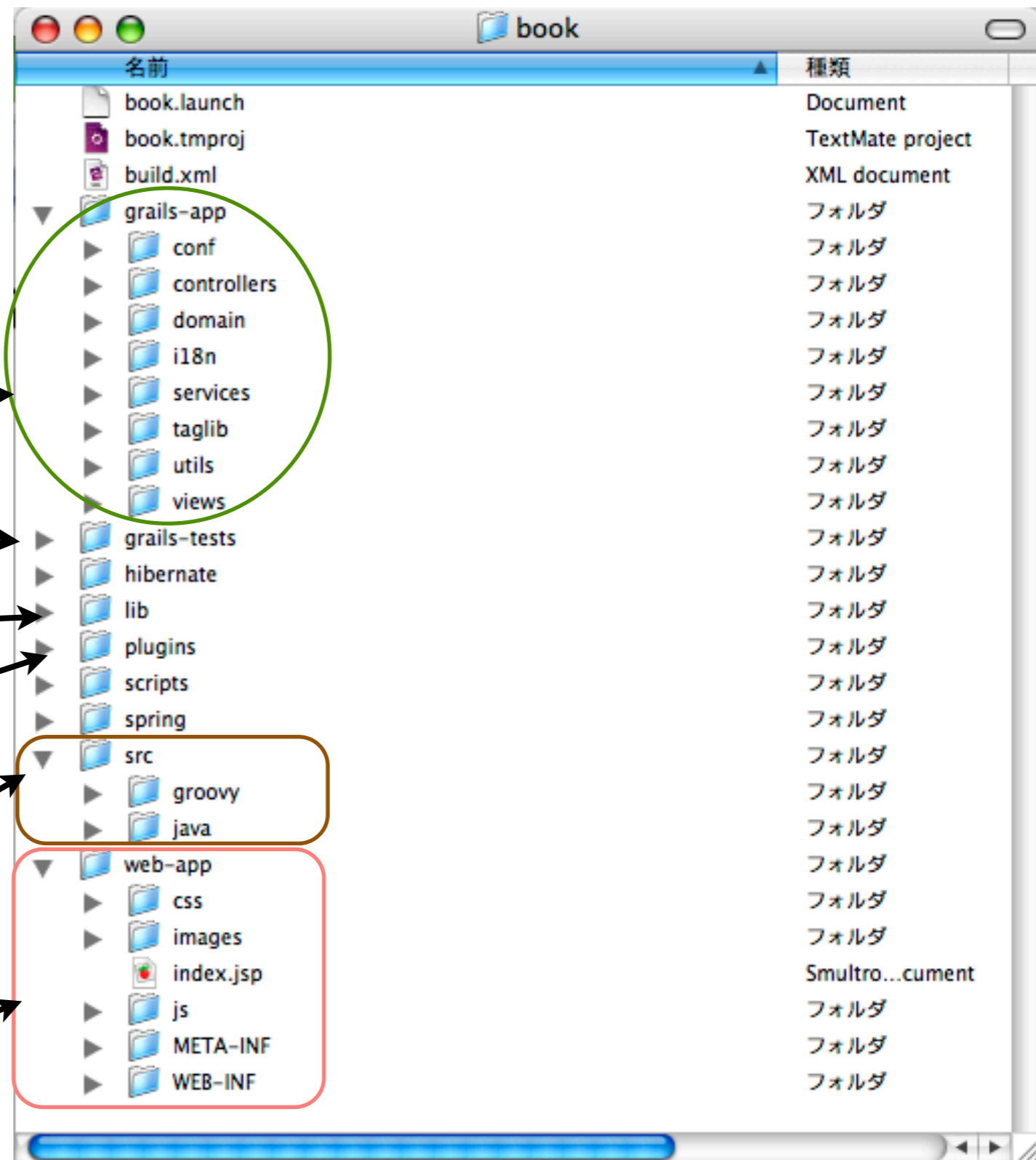
Unit Test用

ライブラリ用

プラグイン用

追加ソース用

css,javascript,等





Webアプリケーション開発の流れ

★Model

処理の中核を担う。

ドメインクラスの作成

★View

表示・出力を司る。

gsp,html,cssで表示を作成。

★Controller

入力を受け取ってその内容に応じてViewとModelを制御する

*Controller.groovyで入力情報を制御。



ドメインクラスを作成

- 永続化データを定義するために、ドメインクラスを作成します。
- **ドメインクラスって何???**
オブジェクト・リレーショナル・マッピングを行うための永続オブジェクトです。
よく分からない場合は、DBテーブルレコードのデータの定義と考えてください。



ドメインクラスを作成

```
# grails create-domain-class book
```

コマンドを実行すると空のクラスファイルが生成されます。

```
class Book {  
  
}
```

フィールド、
バリデーションなどを
書き込む

```
class Book {  
    String title  
    String author  
    String isbn  
  
    static constraints = {  
        title(size:1..120,blank:false)  
        author(size:1..120,blank:false)  
        isbn(blank:false)  
    }  
}
```



データベースの設定

開発時はメモリー上で動作する“HSQLDB”をデータベースとして使用します。

定義ファイルを変更することによって、別のデータベースにも対応できます。

```
class DevelopmentDataSource {
```

DBプーリング設定

```
boolean pooling = true
```

テーブル自動生成の指定

```
String dbCreate = "create-drop" // one of 'create', 'create-drop', 'update'
```

```
String url = "jdbc:hsqldb:mem:devDB"
```

JDBCのURL

```
String driverClassName = "org.hsqldb.jdbcDriver"
```

```
String username = "sa"
```

ドライバークラス名

```
String password = ""
```

```
}
```



対応データベース

Grailsは、Hibernateで対応しているデータベースに対応しています。

DB2
PostgreSQL
MySQL (InnoDB, MyISAM)
Oracle
Sybase
Microsoft SQL Server
SAP DB
Informix
HypersonicSQL
Ingres
Pointbase
FrontBase
他、



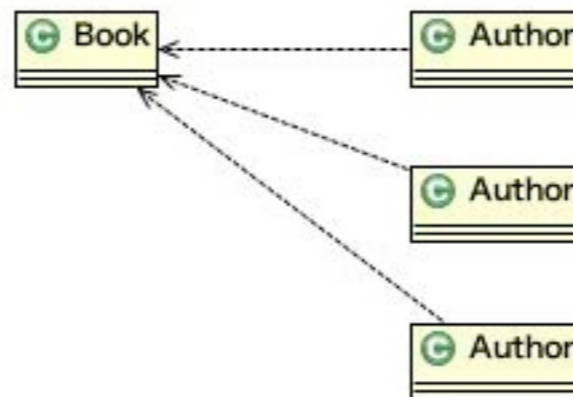
GORM (Grails ORM)

◆ リレーションのサポート

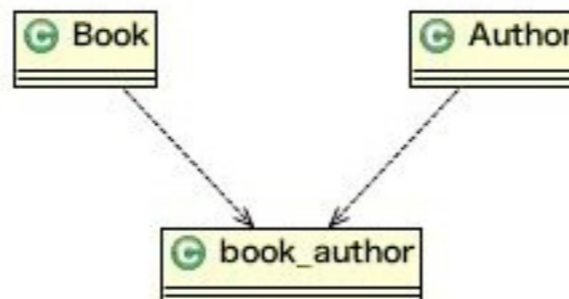
1 : 1 - One to One



1 : n - One to Many



m : n - Many to Many





GORM (Grails ORM)

◆ ダイナミックメソッドでのデータアクセス

save() - データの保存

delete() - データの削除

update() - データの更新

list() - データリスト取得

get(**n**) - IDを指定してデータ取得

findAllBy**フィールド名** - 条件で検索

等

```
//リスト取得  
if(!params.max)params.max = 10  
Book.list( params )
```

```
//IDで取得  
def book = Book.get( params.id )
```

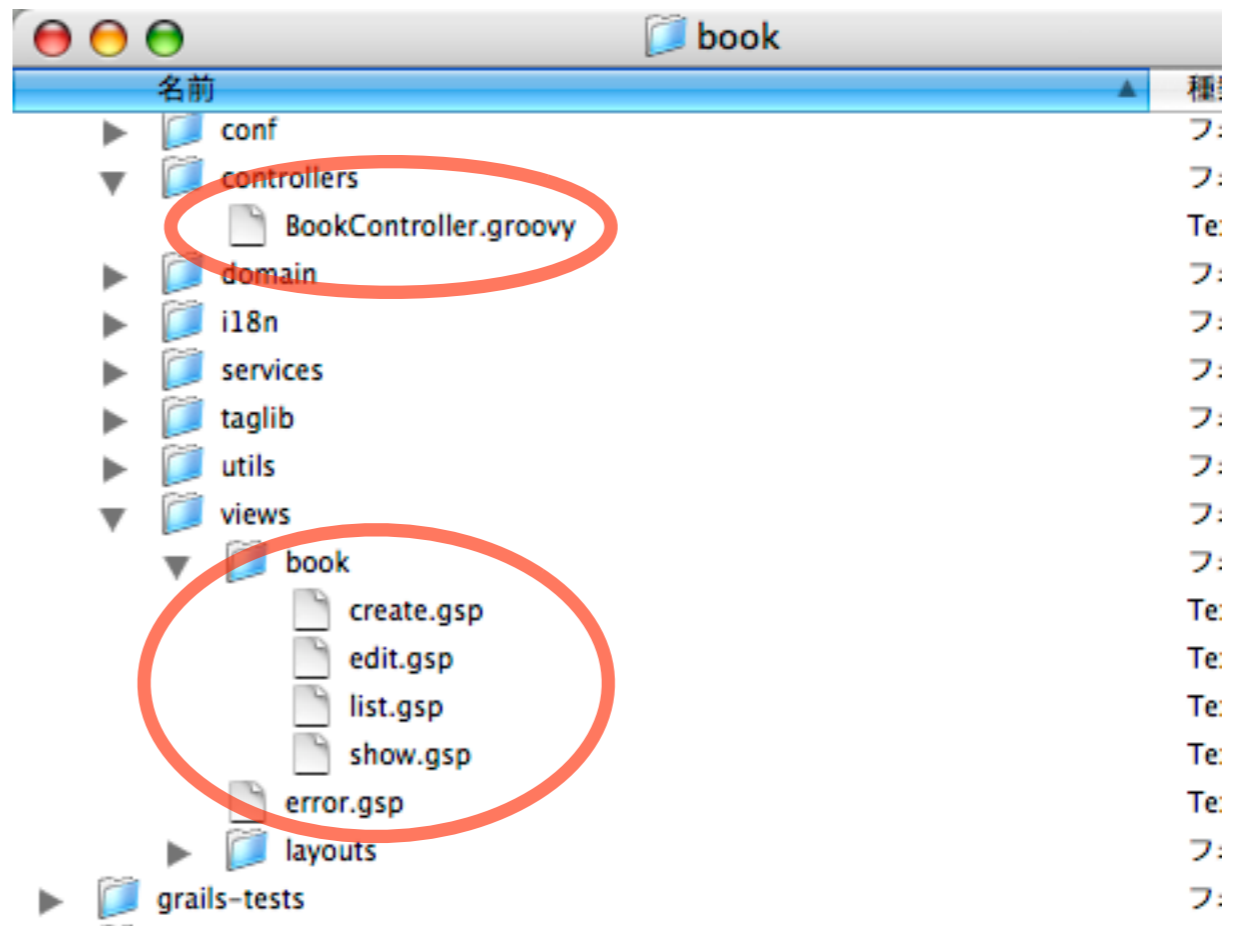
```
//保存  
def book = new Book()  
book.properties = params  
if(book.save()) {  
    redirect(action:show,id:book.id)  
}
```



CRUD : 表示とコントローラ

```
# grails generate-all book
```

コマンドを実行すると
BookController.groovy
viewフォルダ内にbookフォルダ
と中に4つのファイルが生成さ
れます。





Controllerとアクションの関係

URL

コンテキスト (アプリケーション名)

http://127.0.0.1:8080/myapp/book/show/2

```
class BookController {  
  def show = {  
    [ book : Book.get( params.id ) ]  
  }  
  //....  
}
```




Groovy Server Pages (表示の編集)

```
class BookController {  
    def show = {  
        [ book : Book.get( params.id ) ]  
    }  
    //....  
}
```

コントローラ内のアクションクロージャから返させる変数をgspファイルで呼出して表示

view/book/show.gspの一部

```
<tr class="prop">  
    <td valign="top" class="name">Title:</td>  
    <td valign="top" class="value">${book.title}</td>  
</tr>
```



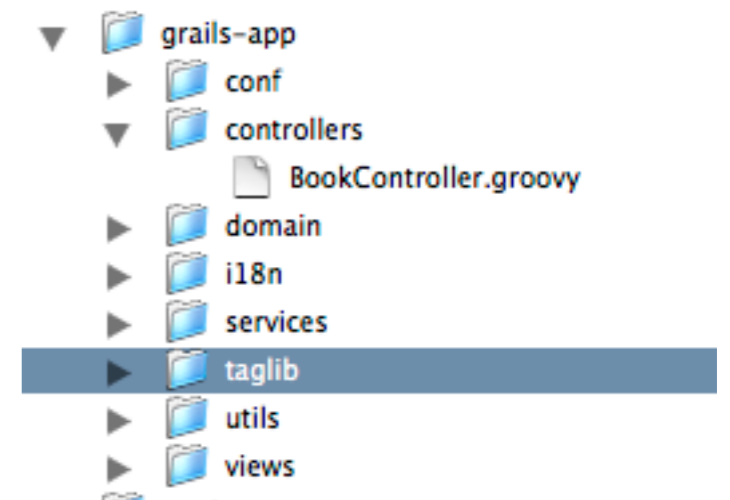
タグライブラリ

- ◆ カスタムタグライブラリを作成することが出来ます。

grails-app/taglibフォルダ内に
TagLibで終わるファイルを作成。

(Ex. SomeTagLib.groovy)

又は、 `grails create-tag-lib some` で作成



- ◆ もちろん、基本タグライブラリも存在します。

`ApplicationTagLib.groovy` `FormTagLib.groovy` `JavascriptTagLib.groovy`

`RenderTagLib.groovy` `UITagLib.groovy` `ValidationTagLib.groovy`



Java統合

- シームレスに既存のJavaライブラリが活用できる。
- JEEアプリケーションサーバへのWARのデプロイ
- Javaで書いたモデルのHibernate定義による利用
- Springで依存注入。



Spring統合

- "\$PROJECT_HOME/spring/resources.xml"へのエントリーの追加することによって、Beansはauto-wiringとして設定することができます。

```
<bean id="myBean" class="MyBeanImpl"></bean>
```



beanのidで定義されたエントリーは、プロパティ名としてGrailsコントローラの中に注入することができます

```
class SomeController {  
    def myBean  
    //.....  
}
```



なぜ、Grailsなのか？

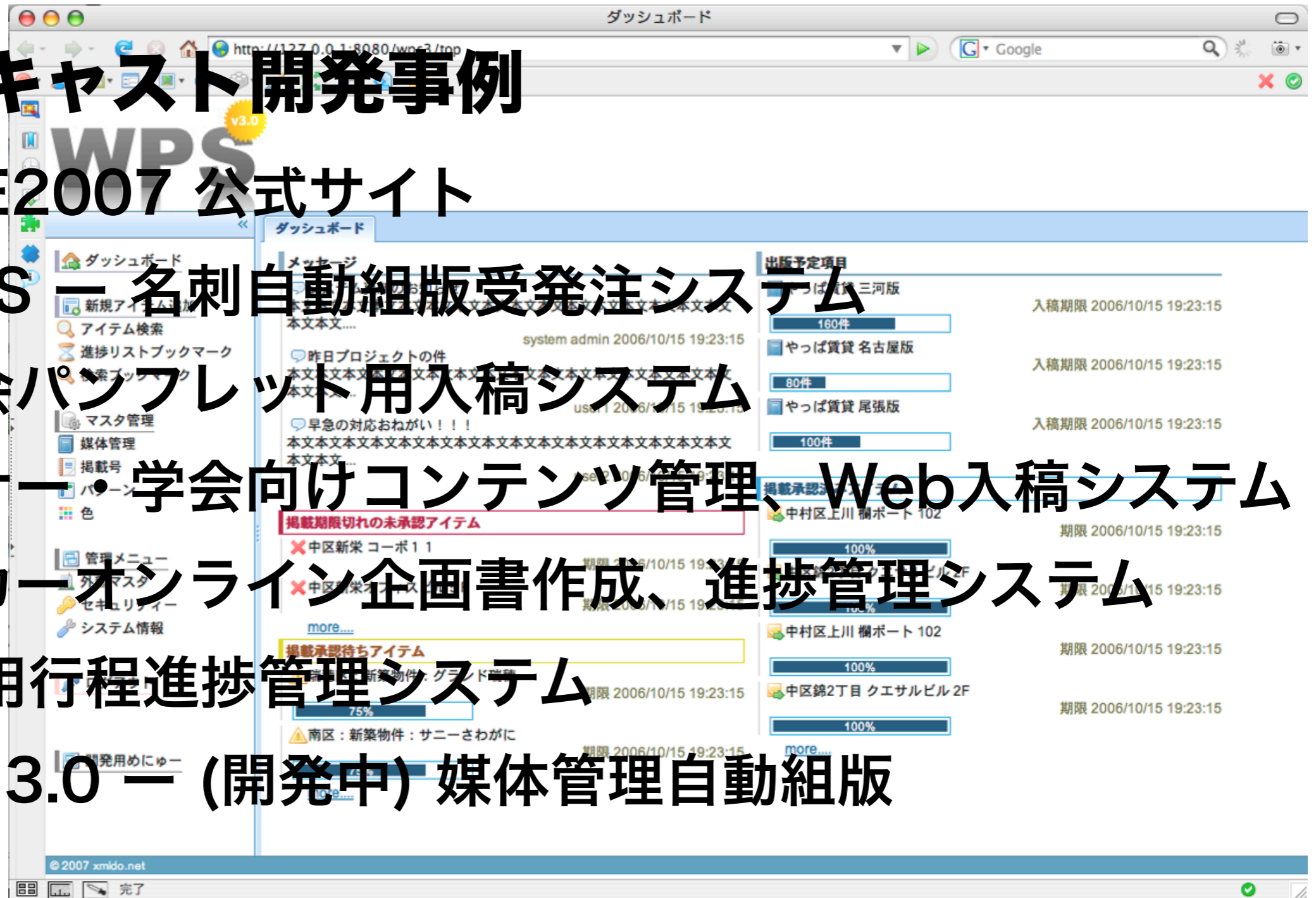
- **Web2.0**の到来。開発成功のキーはアジャイル。
- **Grails**等(RoRやDjango)のダイナミックフレームワークは早い対話式開発で生産性をアップします。
- 一番のポイントは**JVM(Java)**があれば動く！
(マルチプラットフォーム)



開発事例

ニューキャスト開発事例

- PAGE2007 公式サイト
- WBCS 一名刺自動組版受発注システム
- 展示会パンフレット用入稿システム
- セミナー・学会向けコンテンツ管理、Web入稿システム
- メーカーオンライン企画書作成、進捗管理システム
- DTP用行程進捗管理システム
- WPS 3.0 (開発中) 媒体管理自動組版





サイト紹介

- **Grails** – <http://grails.org>
→日本語 <http://grails.org/Japanese+Home>
- **Groovy** – <http://groovy.codehaus.org>
→日本語(翻訳中) <http://groovy.codehaus.org/Japanese+Home>
- **公式メーリングリスト** – <http://grails.org/Mailing+lists>
- **日本のコミュニティー**
<http://groups.google.co.jp/group/grails-ja/>
- **xmlDoブログ** Grails Tips等の情報
<http://xmlDo.blogspot.com/>



(株) ニューキャスト

業務内容：

Web自動組版システム、Webサイト等を開発。
DTP制作。介護事業。

DTP制作現場の声から生まれたノウハウを活用したWeb入稿自動組版などのパッケージ製品を開発販売。

**Grailsを活用したWebアプリケーション開発と
サーバのホスティング。**

Grails用Pluginの開発



Thank You !

