Baking Cakes with CakePHP

박상길 <likejazz@daum.net>

CakePHP

PHP를 이용한 초고속 개발 프레임워크

PHP

- 자바스크립트와 달리 서버 실행 스크립트 언어
- C와 언어 구조 유사, 배우기 쉽고 빠름
- 오픈소스/무료

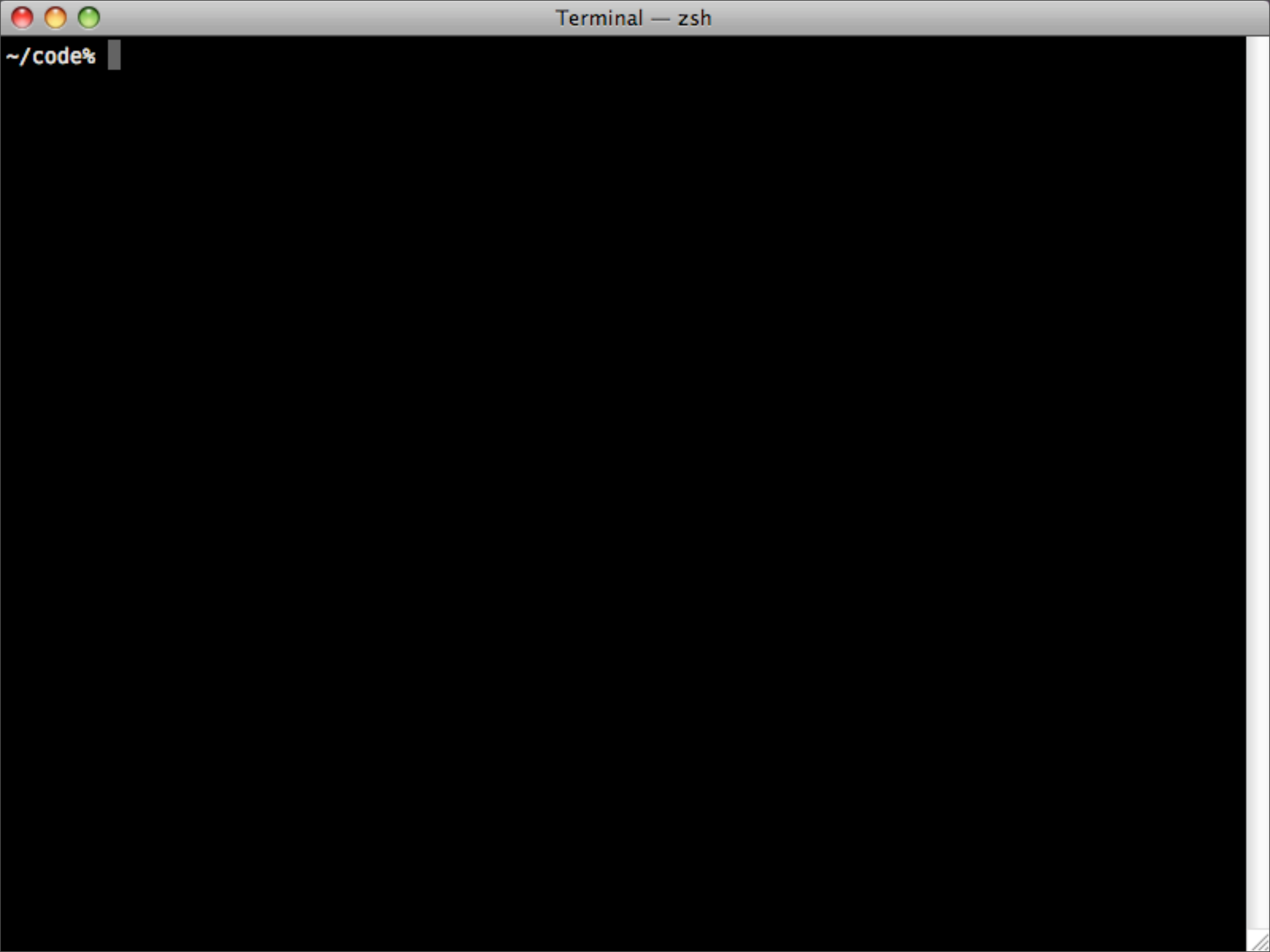
초고속 개발 프레임워크

루비온레일스

- MVC, ORB
- Convention over Configuration
- Scaffolding
- Validation
- Routing
- AJAX, JavaScript, HTML Helpers
- Plugin, Migration

루비온레일스

- 웹 개발에 필요한 툴킷 대부분 탑재
- UI 라이브러리 탑재
- 엄청난 반향



```
~/code%
```

PHP on Rails?

Ruby on Rails!

PHP: 유사 프레임워크 등장

왜 CakePHP 인가

- PHP
- 가장 인기 있음

	Cake	Zend	Symfony	CodeIgniter	eZ Components
Type	Full-stack	Glue / Full-stack	Full-stack	Glue / Full-stack	Glue
PHP 4	Yes	No	No	Yes	No
Code Generation	Yes	No	Yes	No	No
MVC (Push)[1]	Yes, AR	Yes	Yes	Yes, AR	No
ORM	Yes, AR	Yes, Data Gateway	Yes, Doctrine	Yes, Plugin	Yes
AJAX[2]	Yes, Prototype	No	Yes, Prototype	Yes, Plugin	No
I18n	Yes	Yes	Yes	Yes	Yes
Scaffolding	Yes	No	Yes	Yes	No
Unit Testing	Yes, Plugin	Yes	Yes	Yes	Yes
DB Migrations	Yes, Plugin	Yes	Yes, Plugin	Yes, Plugin	Yes
Security	Yes, ACL	Yes, ACL	Yes	Yes	Yes
Template	Yes	Yes	Yes	Yes	Yes
Validation	Yes	Yes	Yes	Yes	Yes
Caching	Yes	Yes	Yes	Yes	Yes

```
./ab –n 500 –c 10 http://localhost/frameworks/baseline.html
                                             /baseline.php
                                             /cake_.../users/
                                             /codeigniter.../index.php/users/
```

Measure	Baseline	Vanilla PHP	CodeIgniter	CakePHP
Requests/sec (mean)	1033.91	166.93	21.48	6.00
Time taken	0.48s	2.99s	23.28s	83.30s
50% reqs max time	15ms	15ms	436ms	936ms

CakePHP 특징

- MVC, ORB
- Convention over Configuration
- Scaffolding
- Validation
- Routing
- AJAX, JavaScript, HTML Helpers
- Plugin, Migration

CakePHP 특징

- 레일스와 매우 유사
- 1.2 버전 이후 독자 노선 모색

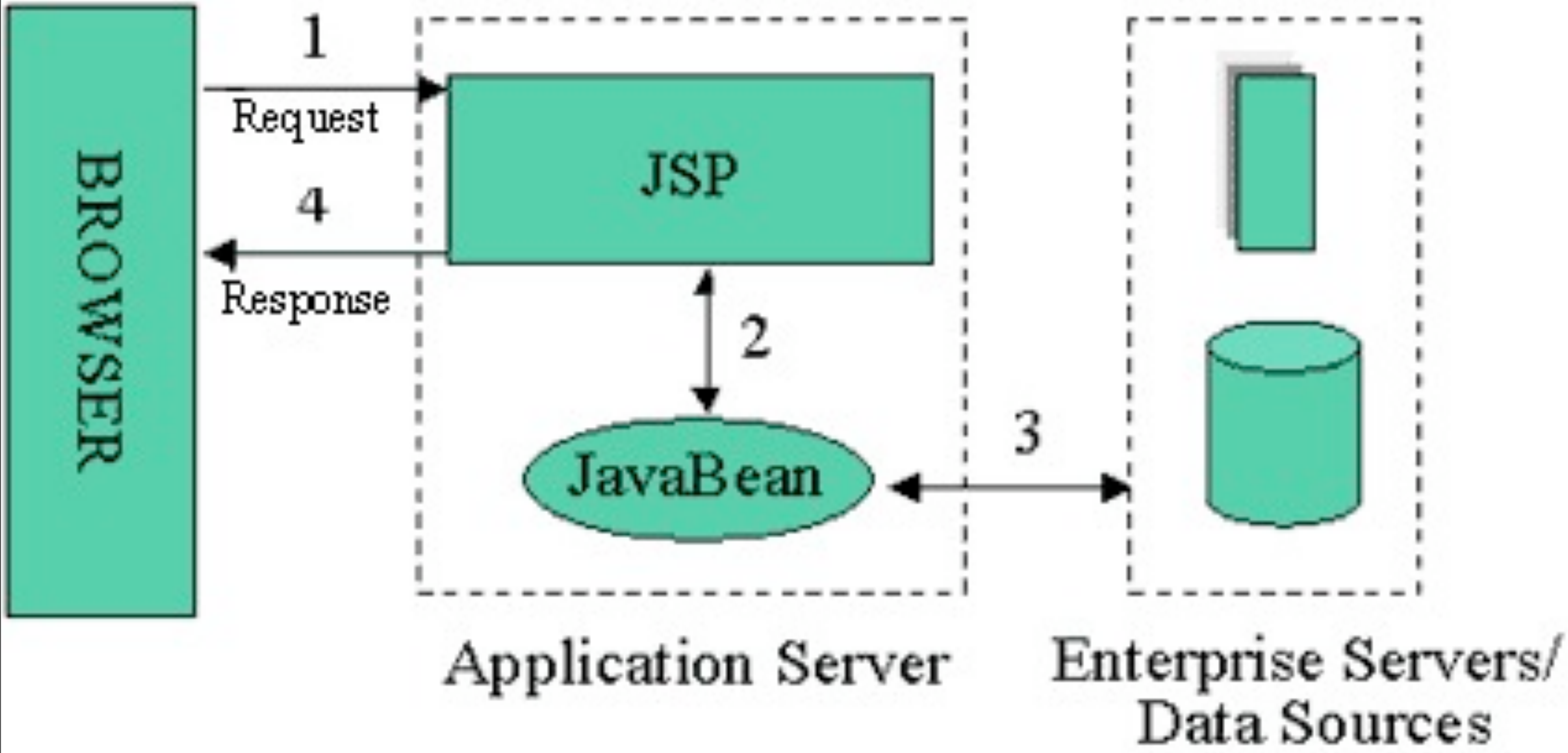
일반적인 PHP 개발 과정

- index 생성
- 각종 라이브러리 작성
- 기능 확장, 정리
- 확장된 기능 정리, 반복

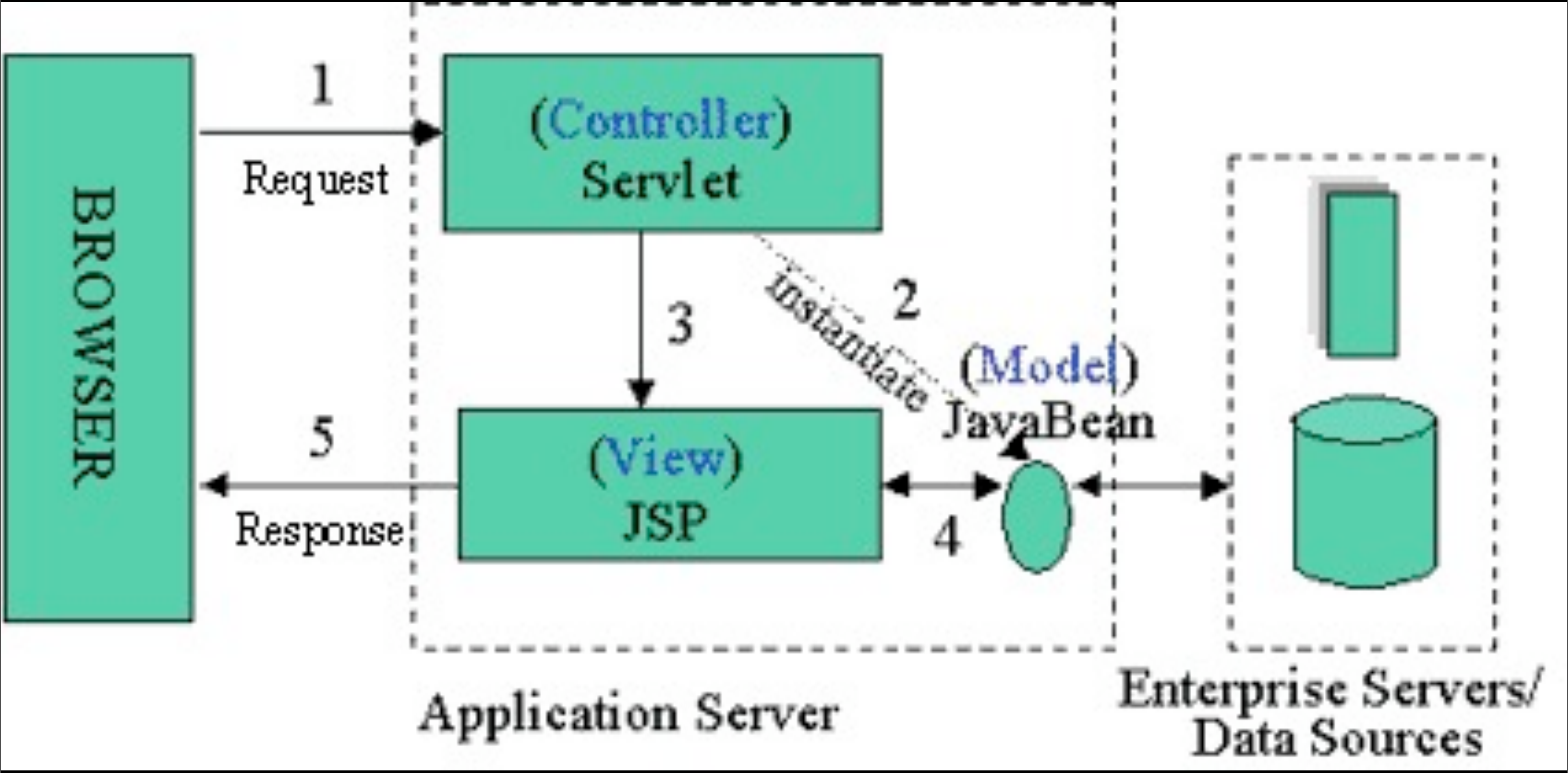
진보적인 PHP 개발 과정

- 데이타 모델 작성
- 스키마 생성
- 데이타 로직 작성
- 비지니스 로직 작성
- 프리젠테이션 레이어 작성

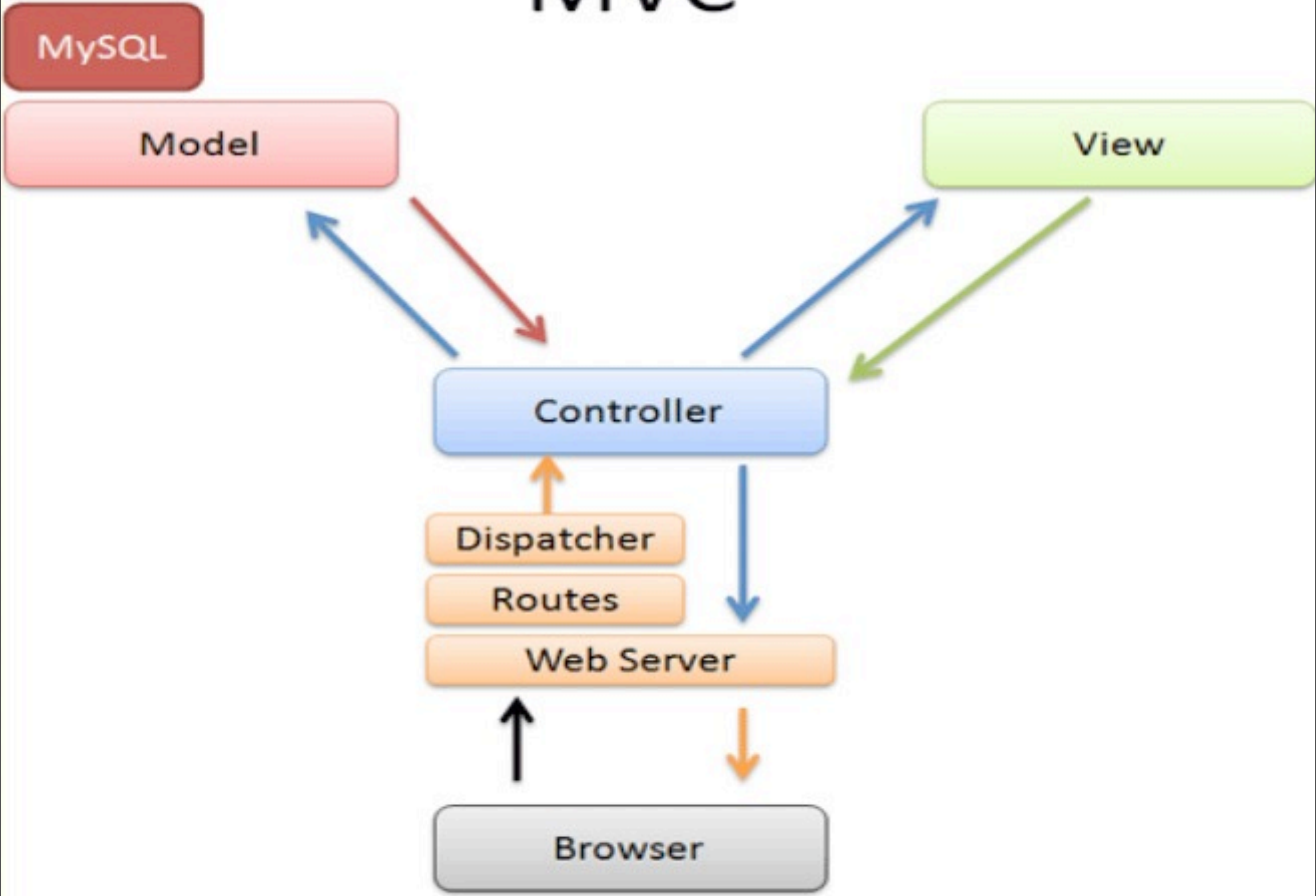
JSP Model 1

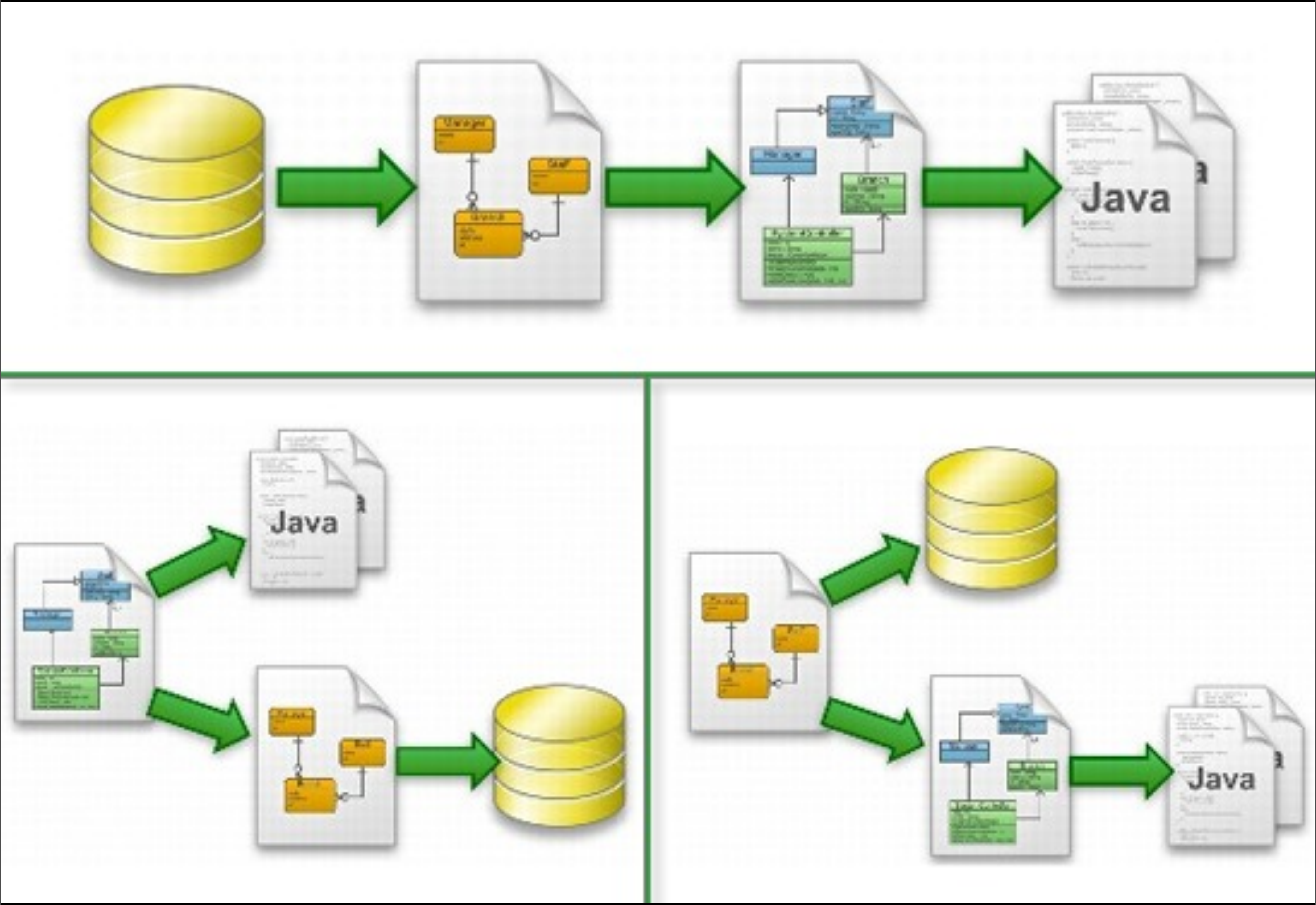

BROWSER
1
Request
4
Response
JSP
2
JavaBean
3
Application Server
Enterprise Servers/
Data Sources

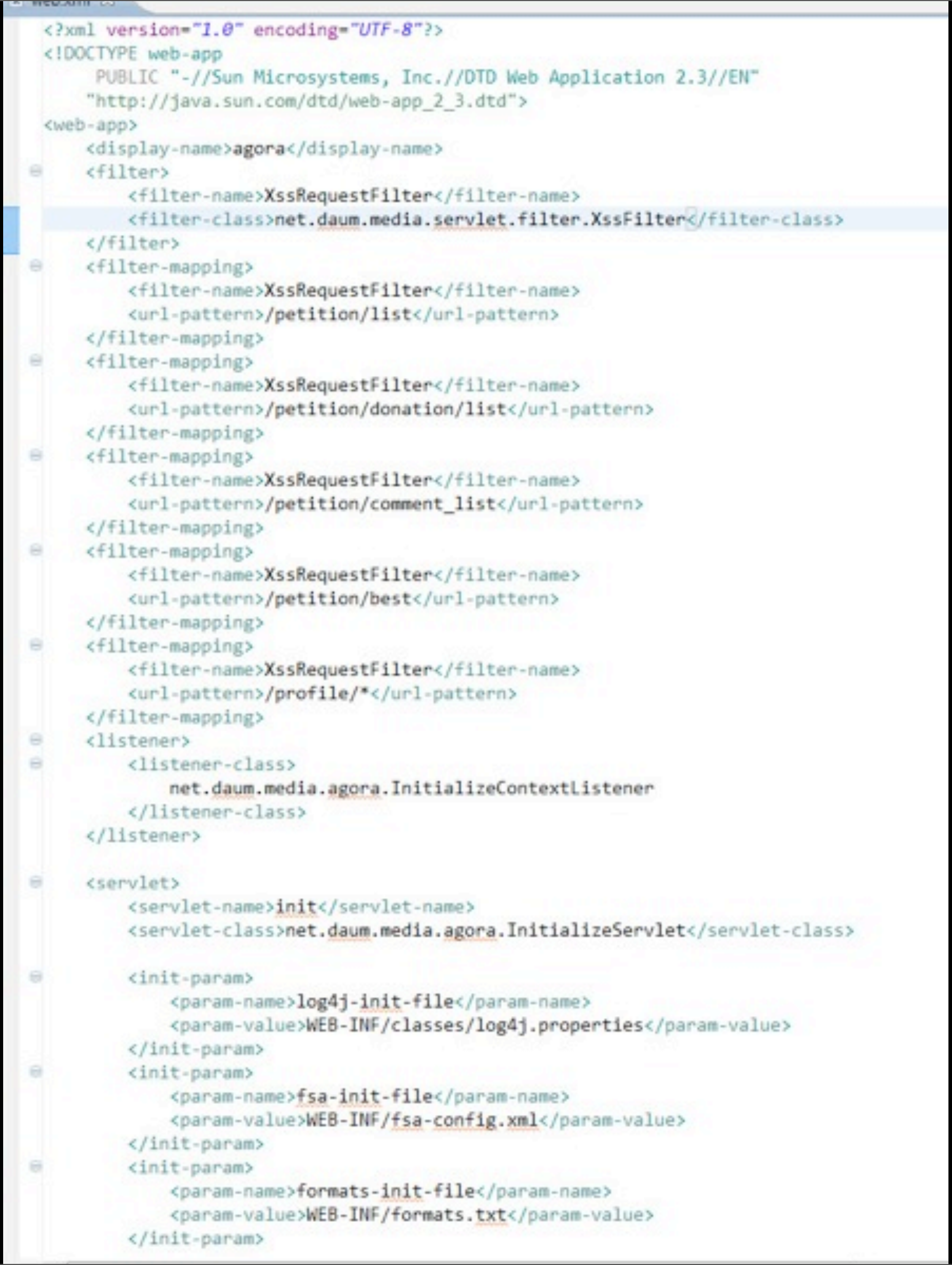
JSP Model 2


BROWSER
1
Request
(Controller)
Servlet
instantiate
2
3
(Model)
JavaBean
5
(View)
JSP
Response
4
Application Server
Enterprise Servers/
Data Sources

MVC


MySQL
Model
View
Controller
Dispatcher
Routes
Web Server
Browser


Java
Java
Java



```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app
     PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
    "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
    <display-name>agora</display-name>
    <filter>
        <filter-name>XssRequestFilter</filter-name>
        <filter-class>net.daum.media.servlet.filter.XssFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>XssRequestFilter</filter-name>
        <url-pattern>/petition/list</url-pattern>
    </filter-mapping>
    <filter-mapping>
        <filter-name>XssRequestFilter</filter-name>
        <url-pattern>/petition/donation/list</url-pattern>
    </filter-mapping>
    <filter-mapping>
        <filter-name>XssRequestFilter</filter-name>
        <url-pattern>/petition/comment_list</url-pattern>
    </filter-mapping>
    <filter-mapping>
        <filter-name>XssRequestFilter</filter-name>
        <url-pattern>/petition/best</url-pattern>
    </filter-mapping>
    <filter-mapping>
        <filter-name>XssRequestFilter</filter-name>
        <url-pattern>/profile/*</url-pattern>
    </filter-mapping>
    <listener>
        <listener-class>
            net.daum.media.agora.InitializeContextListener
        </listener-class>
    </listener>

    <servlet>
        <servlet-name>init</servlet-name>
        <servlet-class>net.daum.media.agora.InitializeServlet</servlet-class>

        <init-param>
            <param-name>log4j-init-file</param-name>
            <param-value>WEB-INF/classes/log4j.properties</param-value>
        </init-param>
        <init-param>
            <param-name>fsa-init-file</param-name>
            <param-value>WEB-INF/fsa-config.xml</param-value>
        </init-param>
        <init-param>
            <param-name>formats-init-file</param-name>
            <param-value>WEB-INF/formats.txt</param-value>
        </init-param>
```

Baking with Cake

```
# 데이타베이스 생성
cake bake app

# database.php 생성
cake bake all User
cake bake all Ingredient
cake bake all Recipe
```

디렉토리 구조

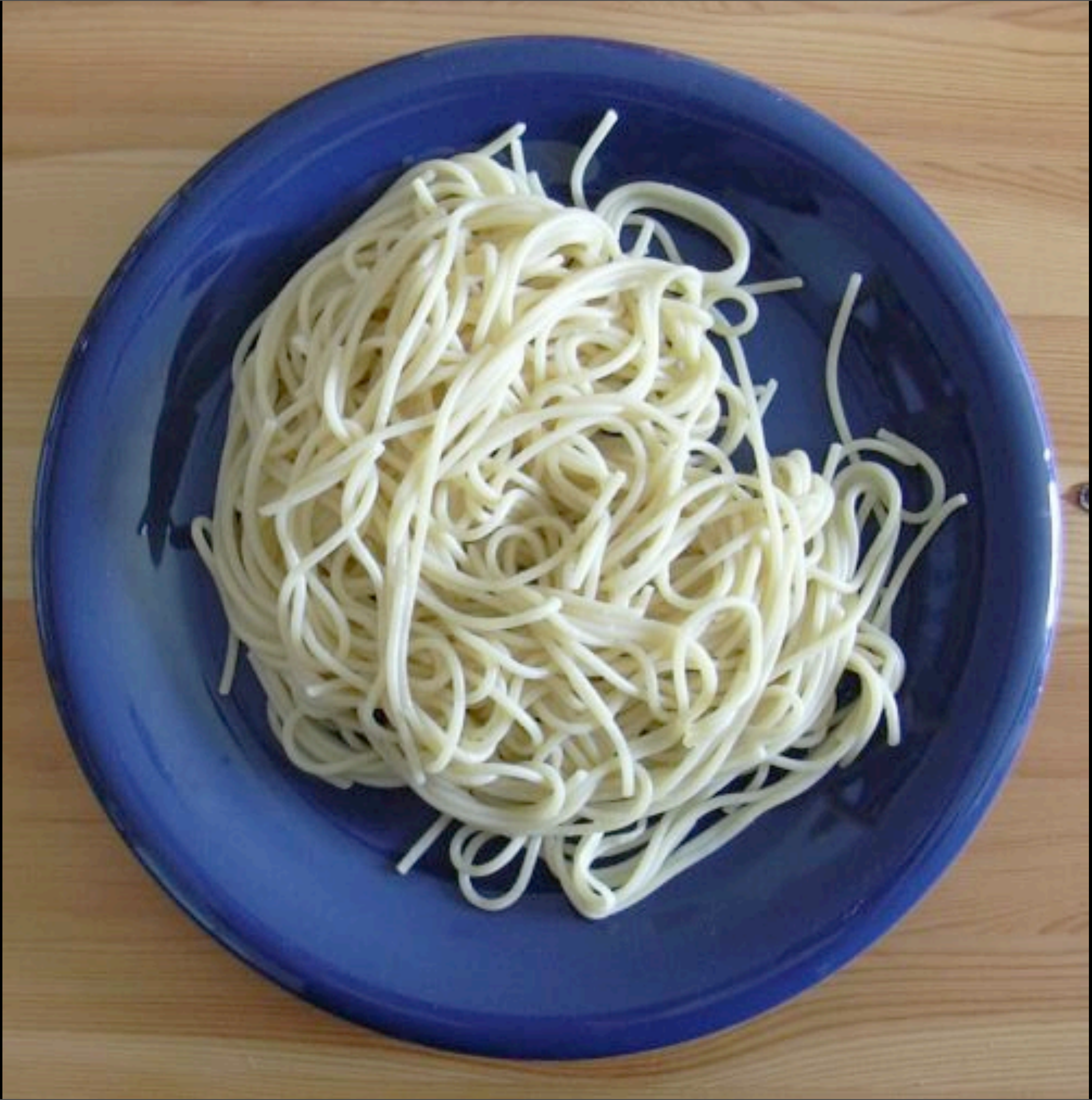
- /
  - app/
    - config/
    - controllers/
    - models/
    - plugins/
    - tmp/
    - vendors/
    - views/
    - webroot/
  - cake
  - docs
  - vendors

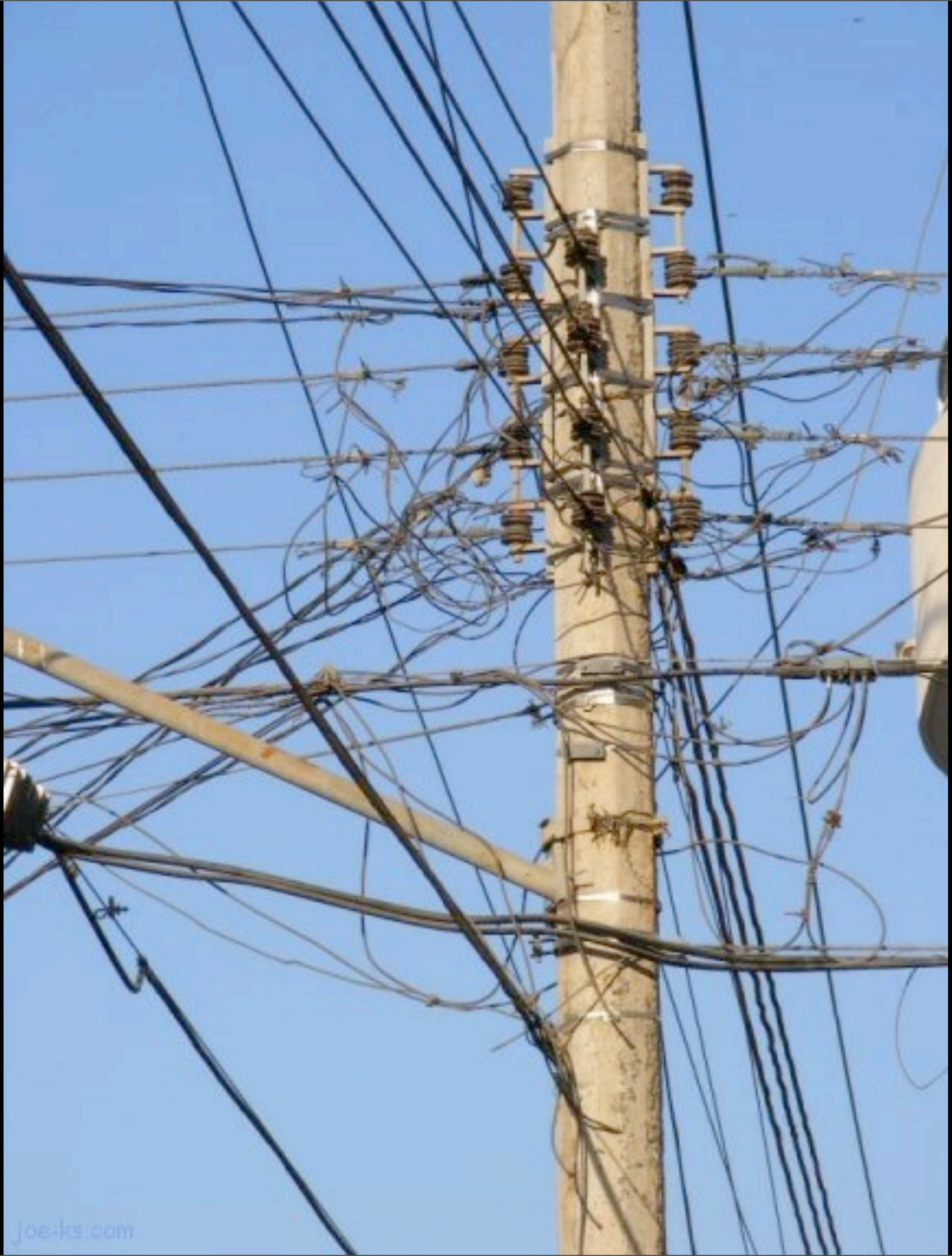
디렉토리 구조

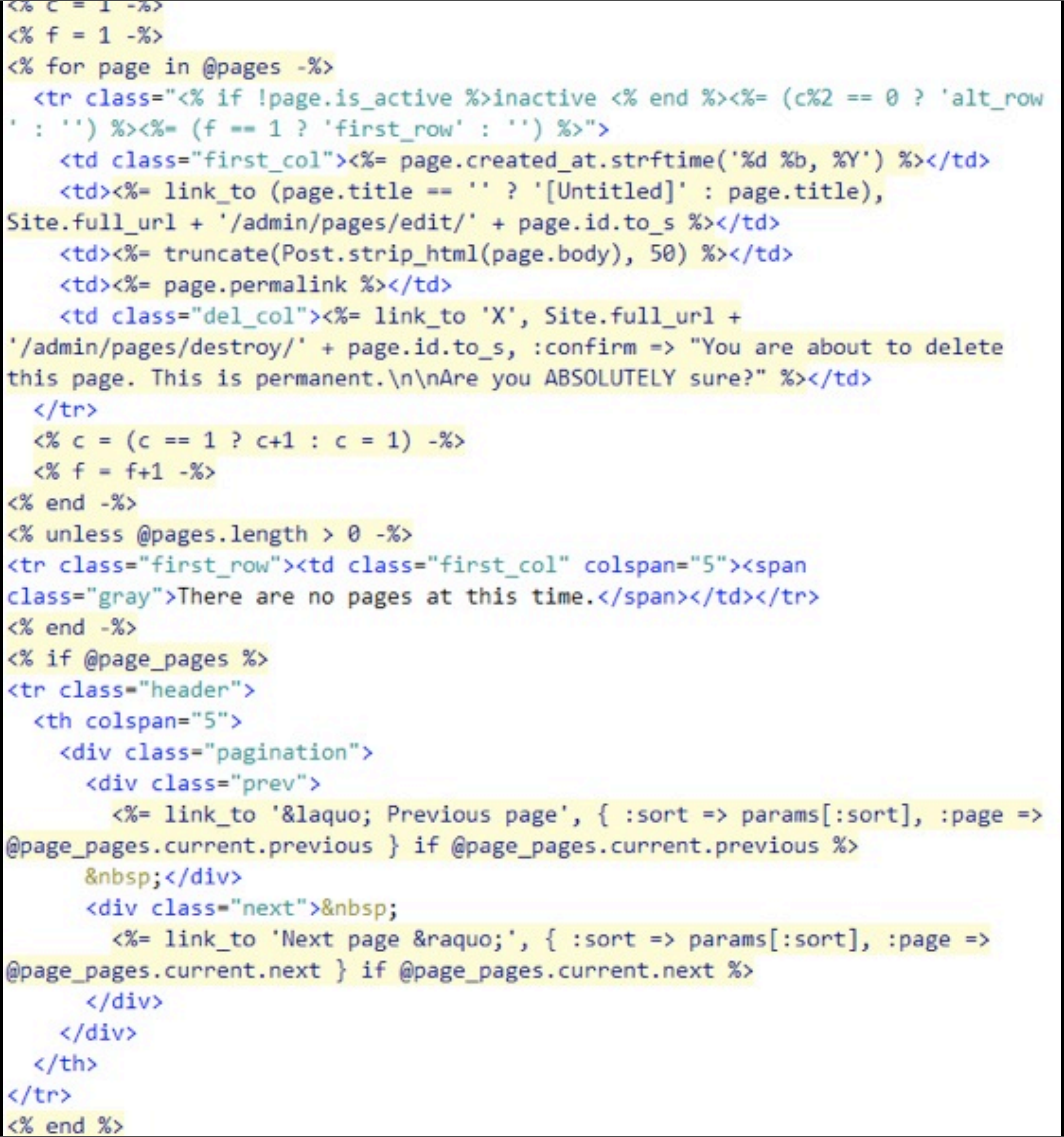
- /
  - app/
    - config/
    - controllers/
    - models/
    - plugins/
    - tmp/
    - vendors/
    - views/
    - webroot/
  - cake
  - docs
  - vendors

익스텐션

- Helpers(for Views)
- Behaviors(for Models)
- Components(for Controllers)

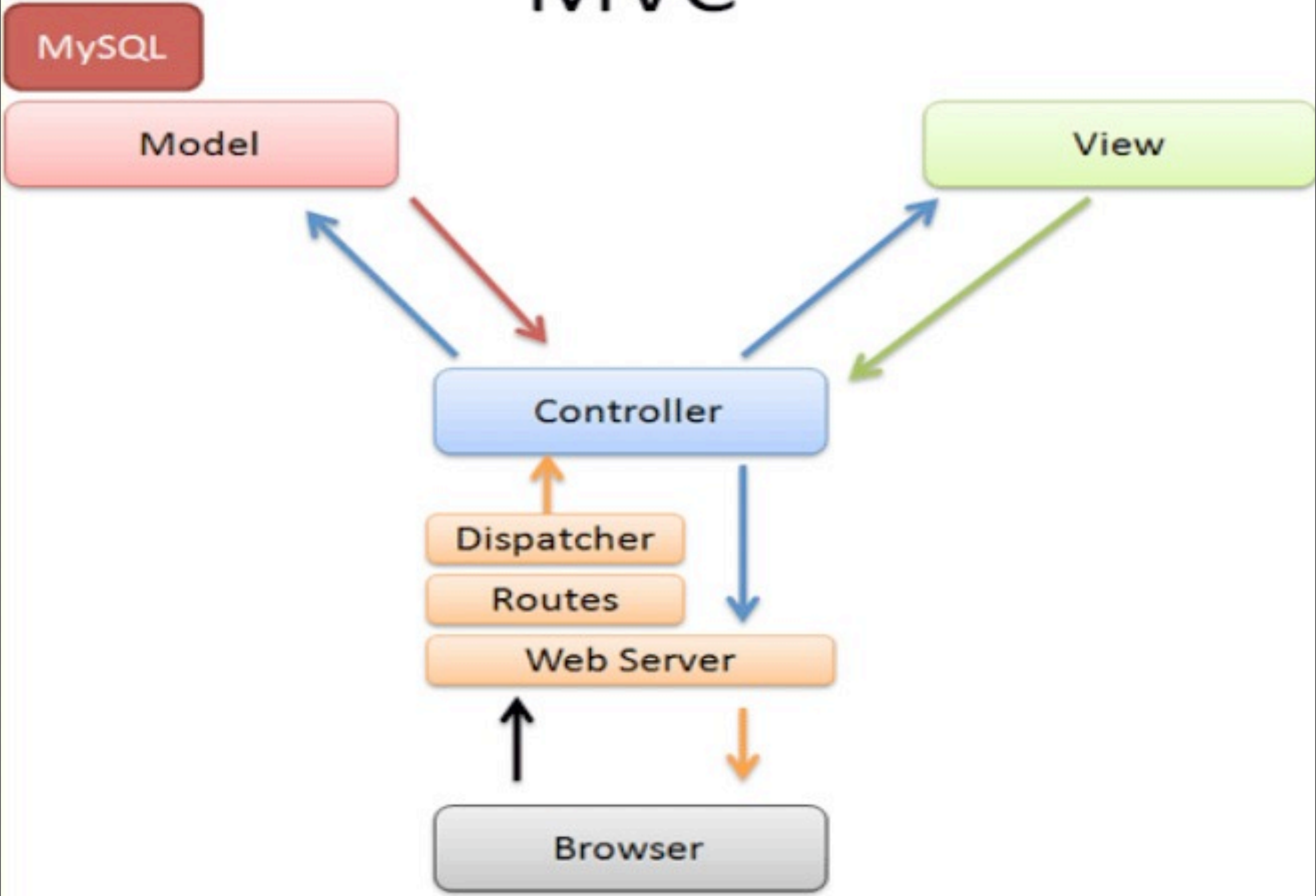



joe-ks.com



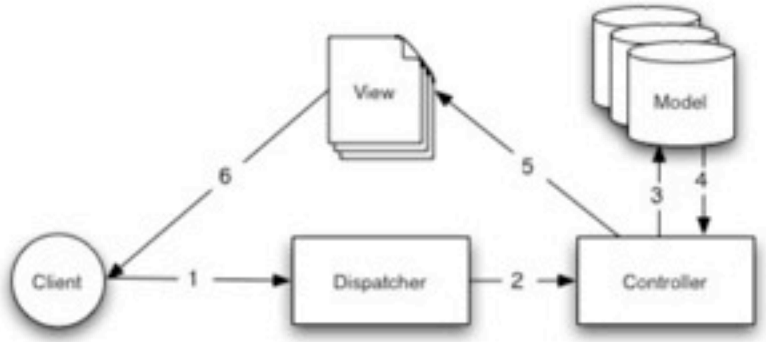
```
<% c = 1 -%>
<% f = 1 -%>
<% for page in @pages -%>
  <tr class="<% if !page.is_active %>inactive <% end %><%= (c%2 == 0 ? 'alt_row
' : '') %><%= (f == 1 ? 'first_row' : '') %>">
    <td class="first_col"><%= page.created_at.strftime('%d %b, %Y') %></td>
    <td><%= link_to (page.title == '' ? '[Untitled]' : page.title),
Site.full_url + '/admin/pages/edit/' + page.id.to_s %></td>
    <td><%= truncate(Post.strip_html(page.body), 50) %></td>
    <td><%= page.permalink %></td>
    <td class="del_col"><%= link_to 'X', Site.full_url +
'/admin/pages/destroy/' + page.id.to_s, :confirm => "You are about to delete
this page. This is permanent.\n\nAre you ABSOLUTELY sure?" %></td>
  </tr>
  <% c = (c == 1 ? c+1 : c = 1) -%>
  <% f = f+1 -%>
<% end -%>
<% unless @pages.length > 0 -%>
<tr class="first_row"><td class="first_col" colspan="5"><span
class="gray">There are no pages at this time.</span></td></tr>
<% end -%>
<% if @page_pages %>
<tr class="header">
  <th colspan="5">
    <div class="pagination">
      <div class="prev">
        <%= link_to '&laquo; Previous page', { :sort => params[:sort], :page =>
@page_pages.current.previous } if @page_pages.current.previous %>
      &nbsp;</div>
      <div class="next">&nbsp;
        <%= link_to 'Next page &raquo;', { :sort => params[:sort], :page =>
@page_pages.current.next } if @page_pages.current.next %>
      </div>
    </div>
  </th>
</tr>
<% end %>
```

MVC


MySQL
Model
View
Controller
Dispatcher
Routes
Web Server
Browser

MVC Example

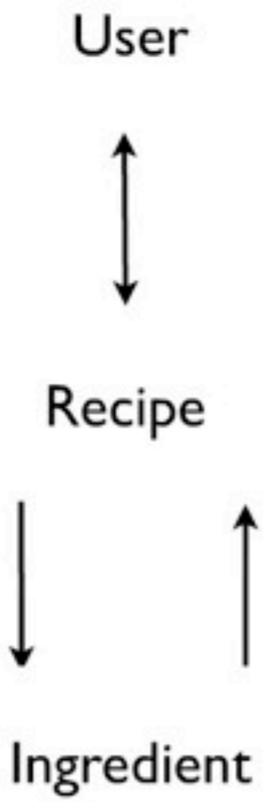
1. Click "Buy a Cake"
2. Dispatcher routes the request to /cake/buy
3. Controller checks if user is logged in, asks ShoppingCart model to add a cake.
4. ShoppingCart model returns the current cart items
5. Cart data provided to shopping cart view.
6. Shopping cart page rendered in User's browser



ORM

- 데이타베이스 테이블 대표
- RDBMS
  - 관계 1:1, 1:n
- CRUD
- 설정 보다 약속

Table Conventions



users

- id
- name

recipes

- id
- name

ingredients

- id
- recipe_id
- name

recipes_users

- recipe_id
- user_id

CRUD: Create, Retrieve, Update, Delete

장점

- 빠른 개발
- 일관된 디자인
- 커뮤니티 기여

단점

- 스케일이 힘듬
- Enterprise를 위한 기능 누락
- 숙련된 개발자 부족

실습

- 윈도우 실습
  - XAMPP
  - WAMP
  - APMSETUP