15장. MVC Model

강사 강태광

1.JSP와 서블릿 기반의 설계 모델 – 모델 1과 모델 2

1. JSP 규격서의 초기 버전에서 소개하고 있는 설계 모델(design model)인 모델 1과 모델 2는 웹 애플리케이션이 해야 할 일을 다음 세가지로 구분하여 모듈화
   - 데이터 입력
   - 데이터 처리
   - 데이터 출력

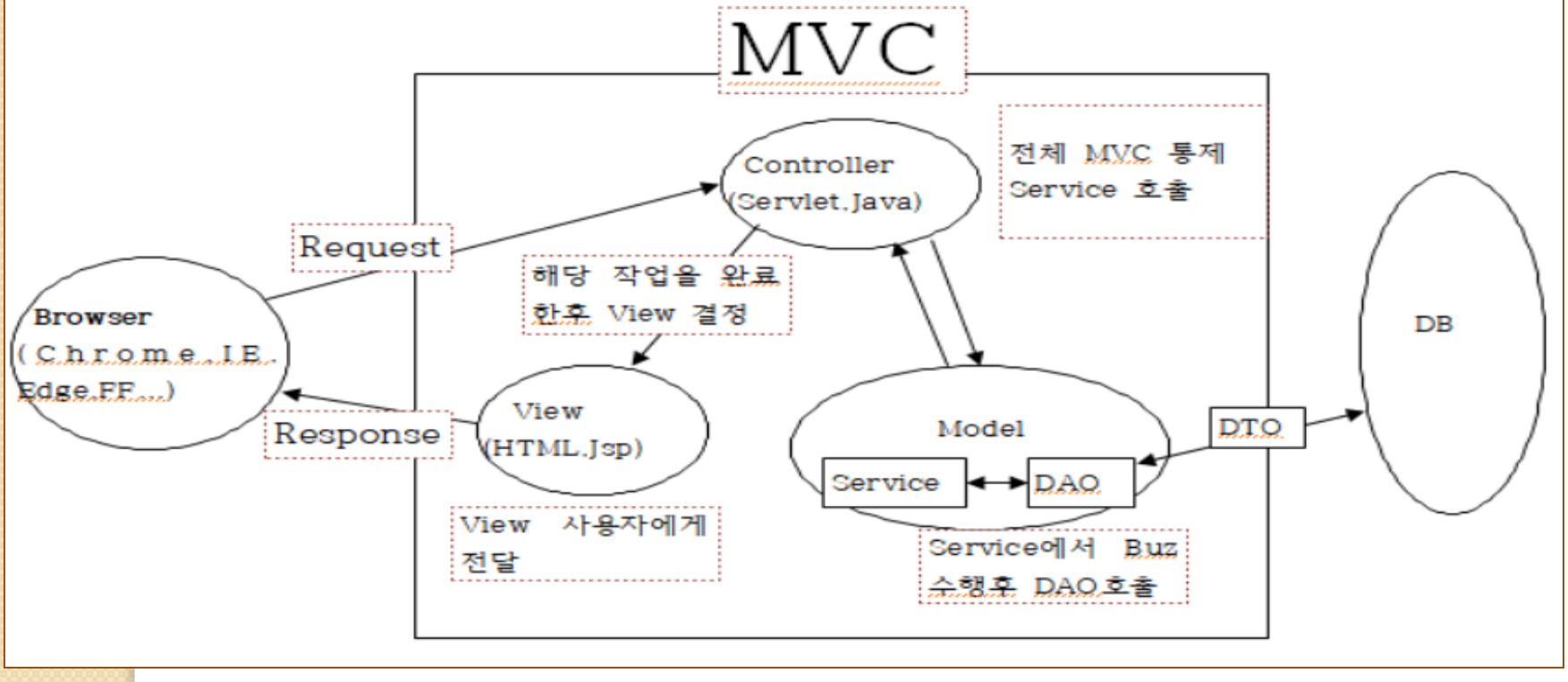
2. 모델 1은 비교적 간단한 웹 애플리케이션에 적합한 설계 모델이고, 모델 2는 비교적 복잡한 웹 애플리케이션에 적합한 설계 모델

3. 모델1(Model 1)구조에서는, 웹 브라우저의 요청(request)을 받아들이고 , 웹 브라우저에 응답(response) 해주는 처리에 대해 JSP page 단독으로 처리하는 구조

2. 모델2(Model 2)

- 모델2(Model 2)구조에서는 요청(request)처리, 데이터접근(data access), 비즈니스 로직(business logic)을 포함하고 있는 컨트롤 컴포넌트(control component) 와 뷰 컴포넌트(view component)는 엄격히 구분

- MVC Pattern에 대해 개념도 및 MVC2 Model에서의 Model , View , Controller의 역할



3.MVC패턴(Model-View-Controller pattern)

1. MVC(Model-View-Controller) 구조는 전통적인 GUI(Graphic User interface) 기반의 어플리케이션을 구현하기 위한 Architecture Style.
   MVC 구조는 사용자의 입력을 받아서, 그 입력 혹은 이벤트에 대한 처리를 하고, 그 결과를 다시 사용자에게 표시하기 위한 최적화된 설계를 제시.
2. 뷰(View)는 화면에 내용을 표출하는 역할을 담당하는 것으로 데이터가 어떻게 생성되고, 어디서 왔는지에 전혀 관여하지 않음.
   단지 정보를 보여주는 역할만을 담당
   JSP기반의 웹 어플리케이션에서는 JSP페이지가 뷰(View)에 해당
3. 컨트롤러(Controller)는 어플리케이션의 흐름을 제어하는 것으로 뷰(View)와 모델(Model)사이에서 이들의 흐름을 제어.
   컨트롤러(Controller)는 사용자의 요청을 받아서 모델(Model)에 넘겨주고, 모델(Model)이 처리한 작업의 결과를 뷰(View)에 보내주는 역할.
   JSP기반의 웹 어플리케이션에서는 보통 서블릿(Servlet)을 컨트롤러 (Controller)로 사용

4.-1 OWASP (Open Web Application Security Project)

1. 악용가능성, 탐지가능성 및 영향에 대해 빈도수가 높고 보안상 영향을 크게 줄 수 있는 10가지 웹 애플리케이션 보안 취약점 목록.

2. OWASP Top 10 목록은 3~4년에 한번씩 정기적으로 업데이트

▣ OWASP TOP10 2021 주요 내용

A01 : Broken Access Control (접근 권한 취약점)

엑세스 제어는 사용자가 권한을 벗어나 행동할 수 없도록 정책을 시행합니다. 만약 엑세스 제어가 취약하면 사용자는 주어진 권한을 벗어나 모든 데이터를 무단으로 열람, 수정 혹은 삭제 등의 행위로 이어질 수 있습니다.

Ex : CWE-352: Cross-Site Request Forgery

A02 : Cryptographic Failures (암호화 오류)

Sensitive Data Exposure(민감 데이터 노출)의 명칭이 2021년 Cryptographic Failures(암호화 오류)로 변경되었습니다. 적절한 암호화가 이루어지지 않으면 민감 데이터가 노출될 수 있습니다.

Ex : CWE-259: Use of Hard-coded Password

A03: Injection (인젝션)

SQL, NoSQL, OS 명령, ORM(Object Relational Mapping), LDAP, EL(Expression Language) 또는 OGNL(Object Graph Navigation Library) 인젝션 취약점은 신뢰할 수 없는 데이터가 명령어나 쿼리문의 일부분으로써, 인터프리터로 보내질 때 취약점이 발생합니다.

Ex : CWE-79: Cross-site Scripting

A04: Insecure Design (안전하지 않은 설계)

Insecure Design(안전하지 않은 설계)는 누락되거나 비효율적인 제어 설계로 표현되는 다양한 취약점을 나타내는 카테고리 입니다. 안전하지 않은 설계와 안전하지 않은 구현에는 차이가 있지만, 안전하지 않은 설계에서 취약점으로 이어지는 구현 결함이 있을 수 있습니다.

Ex : CWE-209(중요한 정보가 포함된 오류 메시지 생성), CWE-522: 불충분하게 보호된 자격 증명

4-2. OWASP (Open Web Application Security Project)

1. 악용가능성, 탐지가능성 및 영향에 대해 빈도수가 높고 보안상 영향을 크게 줄 수 있는 10가지 웹 애플리케이션 보안 취약점 목록.

2. OWASP Top 10 목록은 3~4년에 한번씩 정기적으로 업데이트

▣ OWASP TOP10 2021 주요 내용

A05: Security Misconfiguration (보안설정오류)

애플리케이션 스택의 적절한 보안 강화가 누락되었거나 클라우드 서비스에 대한 권한이 적절하지 않게 구성되었을 때, 불필요한 기능이 활성화 되거나 설치되었을 때, 기본계정 및 암호화가 변경되지 않았을 때, 지나치게 상세한 오류 메세지를 노출할 때, 최신 보안기능이 비활성화 되거나 안전하지 않게 구성되었을 때 발생합니다.

Ex : CWE-611 Improper Restriction of XML External Entity Reference

A06: Vulnerable and Outdated Components (취약하고 오래된 요소)

취약하고 오래된 요소는 지원이 종료되었거나 오래된 버전을 사용할 때 발생합니다. 이는 애플리케이션 뿐만 아니라, DBMS, API 및 모든 구성요소 들이 포함됩니다.

Ex : CWE-1104: Use of Unmaintained Third-Party Components

A07: Identification and Authentication Failures (식별 및 인증 오류)

Broken Authentication(취약한 인증)으로 알려졌던 해당 취약점은 identification failures(식별 실패)까지 포함하여 더 넓은 범위를 포함할 수 있도록 변경되었습니다. 사용자의 신원확인, 인증 및 세션관리가 적절히 되지 않을 때 취약점이 발생할 수 있습니다.

Ex : CWE-287: Improper Authentication

A08: Software and Data Integrity Failures(소프트웨어 및 데이터 무결성 오류)

2021년 새로 등장한 카테고리로 무결성을 확인하지 않고 소프트웨어 업데이트, 중요 데이터 및 CI/CD 파이프라인과 관련된 가정을 하는데 중점을 둡니다.

Ex : CWE-502: Deserialization of Untrusted Data

4-3. OWASP (Open Web Application Security Project)

1. 악용가능성, 탐지가능성 및 영향에 대해 빈도수가 높고 보안상 영향을 크게 줄 수 있는 10가지 웹 애플리케이션 보안 취약점 목록.

2. OWASP Top 10 목록은 3~4년에 한번씩 정기적으로 업데이트

▣ OWASP TOP10 2021 주요 내용

A09: Security Logging and Monitoring Failures (보안 로깅 및 모니터링 실패)

Insufficient Logging & Monitoring(불충분한 로깅 및 모니터링) 명칭이었던 카테고리가 Security Logging and Monitoring Failures (보안 로깅 및 모니터링 실패)로 변경되었습니다. 로깅 및 모니터링 없이는 공격 활동을 인지할 수 없습니다. 이 카테고리는 진행중인 공격을 감지 및 대응하는데 도움이 됩니다.
Ex : CWE-778 Insufficient Logging

A10: Server-Side Request Forgery (서버 측 요청 위조)

2021년 새롭게 등장하였습니다. SSRF 결함은 웹 애플리케이션이 사용자가 제공한 URL의 유효성을 검사하지 않고 원격 리소스를 가져올 때마다 발생합니다. 이를 통해 공격자는 방화벽, VPN 또는 다른 유형의 네트워크 ACL(액세스 제어 목록)에 의해 보호되는 경우에도 응용 프로그램이 조작된 요청을 예기치 않은 대상으로 보내도록 강제할 수 있습니다.

5-1. bootstract 선언

1. https://getbootstrap.com/docs/5.0/getting-started/introduction/ **Site 검색**

- Getting started
  - Introduction
  - Download
  - Contents
  - Browsers & devices
  - JavaScript
  - Build tools
  - Webpack
  - Parcel
  - Accessibility
  - RFS
  - RTL
- Customize
- Layout
- Content
- Forms
- Components
- Helpers
- Utilities
- Extend
- About

Migration

Introduction

View on GitHub

Get started with Bootstrap, the world's most popular framework for building responsive, mobile-first sites, with jsDelivr and a template starter page.

Adobe Creative Cloud for Teams. Put creativity to work.

ads via Carbon

Quick start

Looking to quickly add Bootstrap to your project? Use jsDelivr, a free open source CDN. Using a package manager or need to download the source files? Head to the downloads page.

CSS

Copy-paste the stylesheet <link> into your <head> before all other stylesheets to load our CSS.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet
```

Copy

JS

Many of our components require the use of JavaScript to function. Specifically, they require our own JavaScript plugins and Popper. Place **one of the following** <script>s near the end of your pages, right before the closing </body> tag, to enable them.

5-2. bootstract 사용

1. https://getbootstrap.com/docs/5.0/getting-started/introduction/ **Site 검색**