



라이브 세션-2022.12.16(금)-API 계층

✓ 전달 사항

1 Solution 코드 README.md에 대한 공지

✓ CSR(Client Side Rendering)과 SSR(Server Side Rendering)

1 Rendering이란?

2 Rendering의 대상은?

3 SSR(Server Side Rendering)

4 CSR(Client Side Rendering)

✓ 애플리케이션 실행 로그 살펴보기

✓ Controller 레퍼런스 코드 리뷰

✓ DTO 레퍼런스 코드 리뷰

✓ Controller 실습 코드 리뷰

✓ DTO 실습 코드 리뷰

✓ Spring MVC 주요 컴포넌트 내부 들여다 보기

1 DispatcherServlet

2 HandlerMapping

3 HandlerMappingAdapter

4 ViewResolver

5 View

6 HttpMessageConverter

✓ 트러블 슈팅 사례 공유

1 문제 해결을 잘하는 방법에 대한 PPT부터 보아요!

2 실제 사례에서 문제점을 코드로 확인해 봅시다!

3 IntelliJ에서의 디버깅 방법

4 ★ 꼭 기억해야 될 문장

직접 겪기 전에는 내꺼 아님

겪어 봐도 내가 해결안하면 아직 내꺼 아님

★ 겪어 보고 내가 해결하면 내꺼 ^^

✓ 전달 사항

1 Solution 코드 README.md에 대한 공지

✓ CSR(Client Side Rendering)과 SSR(Server Side Rendering)

1 Rendering이란?

브라우저에 우리가 보는 구글 메인 같은 화면을 만들어 내는 과정

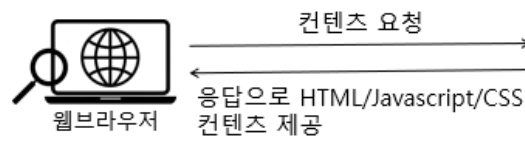
2 Rendering의 대상은?

HTML

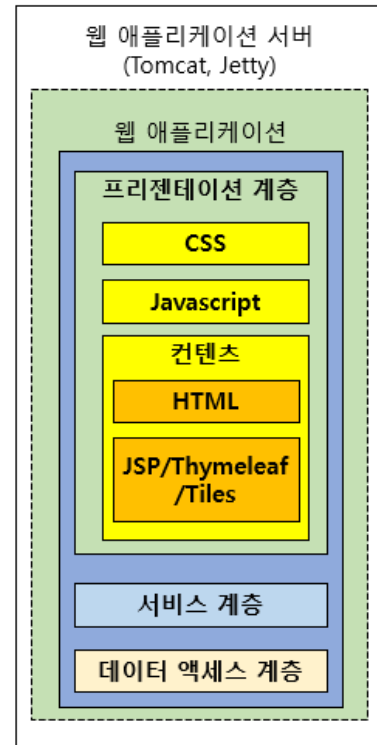
3 SSR(Server Side Rendering)

- 서버 쪽에서 HTML 렌더링
- HTML 페이지를 클라이언트 쪽에 내려준다.
- HTML을 템플릿으로 취급, 동적인 데이터를 채워준다.
- 동기적인 통신
 - 화면이 refresh 된다.

SSR(Server Side Rendering)

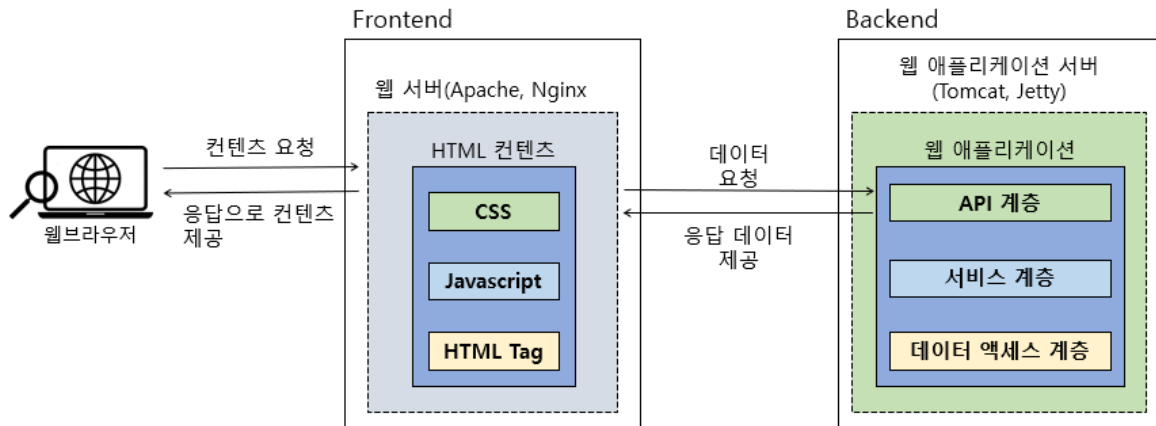


Backend



4 CSR(Client Side Rendering)

CSR(Client Side Rendering)



- Frontend 쪽에서 HTML을 렌더링
- HTML에 포함되는 데이터만 서버에서 받아온다.
 - 대부분 JSON 형태
 - Smile 같은 binary JSON 데이터도
 - <https://github.com/FasterXML/smile-format-specification>
 - 동적인 데이터
- Javascript based 기술로 HTML에 데이터를 채워서 화면을 만든다.
 - SPA(Single Page Application)
 - **React**
 - Vue.js
 - Angular
- **Frontend와 Backend가 분리되어 있는 유형**
 - **Ajax 비동기 통신**
 - 화면이 refresh 되지 않는다.

- Frontend 웹앱 서버
 - Apache
 - Nginx
 - Backend 서버
 - **Java**
 - **Tomcat**
 - **Jetty**
 - ASP.NET
 - IIS
 - PHP
 - Apache
 - Nginx
 - Node.js
 - http 모듈 내장
 - Python
 - http 모듈 내장
-

애플리케이션 실행 로그 살펴보기

- Spring Boot 애플리케이션 실행 후, 출력되는 로그를 분석합니다.
-

Controller 레퍼런스 코드 리뷰

- 코드로 설명합니다.
-

✓ DTO 레퍼런스 코드 리뷰

- 코드로 설명합니다.
-

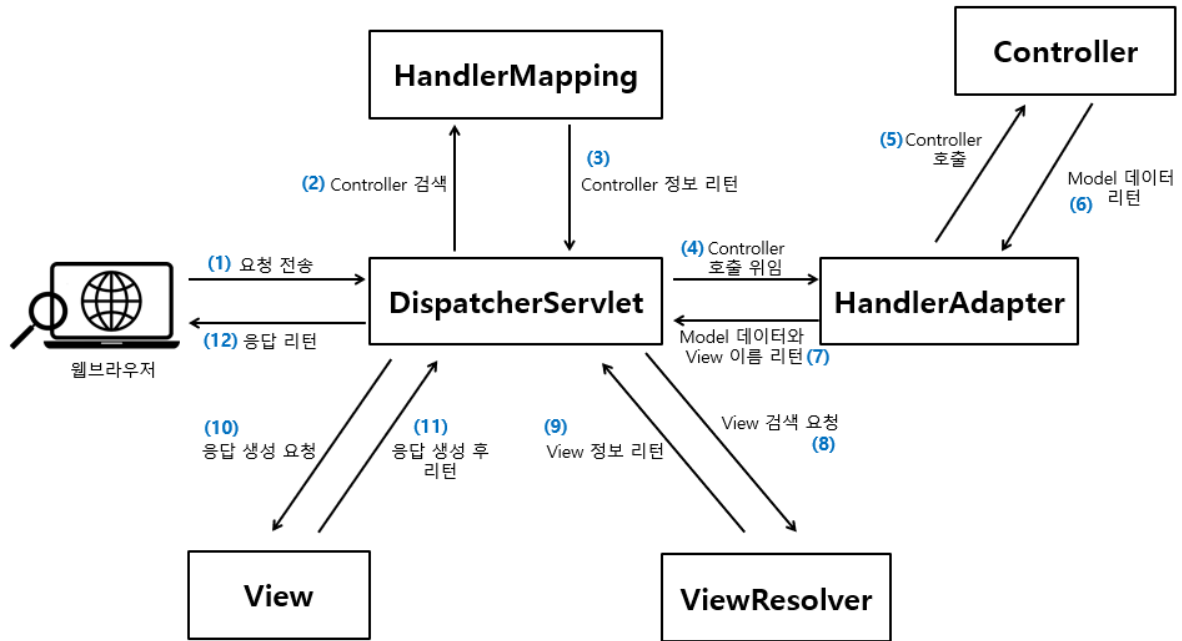
✓ Controller 실습 코드 리뷰

- 코드 설명으로 대체합니다.
-

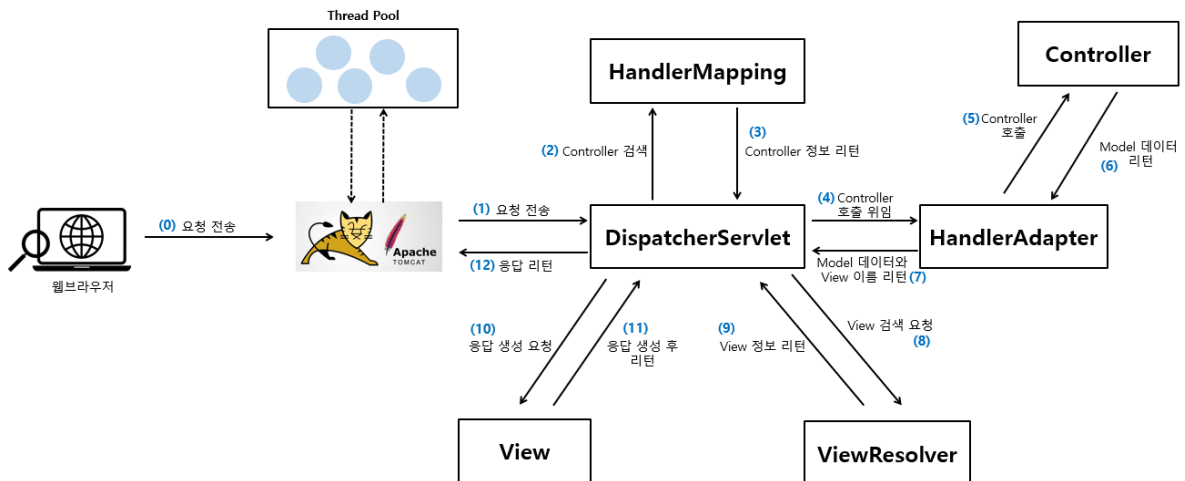
✓ DTO 실습 코드 리뷰

- 코드 설명으로 대체합니다.
-

✓ Spring MVC 주요 컴포넌트 내부 들여다 보기



💡 서블릿 컨테이너가 포함된 모습(one thread per request)



1 DispatcherServlet

- Spring MVC의 Front Controller 역할

2 HandlerMapping

- request와 매치되는 Handler를 찾는다.

3 HandlerMappingAdapter

- Handler를 호출
- `MethodArgumentResolver`를 통해 핸들러에게 파라미터를 넘겨준다.

4 ViewResolver

- 어떤 뷰가 처리할지 결정해준다.

5 View

- 클라이언트에게 전달할 뷰 객체
- 뷰 + 모델 합쳐서 HTML, PDF 등으로 변환해 준다.
- **Rest API는 큰 역할을 안함.**

6 HttpMessageConverter

- **Request Body를 Java 객체로 변환** 해준다.
- **Java 객체를 Response Body로 변환** 해준다.
- **DTO에 setter 메서드가 필요없음**

✓ 트러블 슈팅 사례 공유

1 문제 해결을 잘하는 방법에 대한 PPT부터 보아요!

2 실제 사례에서 문제점을 코드로 확인해 봅시다!

3 IntelliJ에서의 디버깅 방법

- 브레이크 포인트(breakpoint)를 걸어서 디버깅을 해봅시다.

4 ★ 꼭 기억해야 될 문장

직접 겪기 전에는 내꺼 아님

겪어 봐도 내가 해결안하면 아직 내꺼 아님

★ 겪어 보고 내가 해결하면 내꺼 ^^