

8. 다음으로

#2.인강/4. 스프링 MVC 1/강의#

학습 내용 정리

전체 목차

- 1. 웹 애플리케이션 이해
- 2. 서블릿
- 3. 서블릿, JSP, MVC 패턴
- 4. MVC 프레임워크 만들기
- 5. 스프링 MVC - 구조 이해
- 6. 스프링 MVC - 기본 기능
- 7. 스프링 MVC - 웹 페이지 만들기

1. 웹 애플리케이션 이해

- 웹 서버, 웹 애플리케이션 서버
- 서블릿
- 동시 요청 - 멀티 쓰레드
- HTML, HTTP API, CSR, SSR
- 자바 백엔드 웹 기술 역사

2. 서블릿

- 2. 서블릿 - 프로젝트 생성
- 2. 서블릿 - Hello 서블릿
- 2. 서블릿 - HttpServletRequest - 개요
- 2. 서블릿 - HttpServletRequest - 기본 사용법
- 2. 서블릿 - HTTP 요청 데이터 - 개요
- 2. 서블릿 - HTTP 요청 데이터 - GET 쿼리 파라미터
- 2. 서블릿 - HTTP 요청 데이터 - POST HTML Form
- 2. 서블릿 - HTTP 요청 데이터 - API 메시지 바디 - 단순 텍스트
- 2. 서블릿 - HTTP 요청 데이터 - API 메시지 바디 - JSON
- 2. 서블릿 - HttpServletResponse - 기본 사용법
- 2. 서블릿 - HTTP 응답 데이터 - 단순 텍스트, HTML
- 2. 서블릿 - HTTP 응답 데이터 - API JSON

3. 서블릿, JSP, MVC 패턴

- 3. 서블릿, JSP, MVC 패턴 - 회원 관리 웹 애플리케이션 요구사항
- 3. 서블릿, JSP, MVC 패턴 - 서블릿으로 회원 관리 웹 애플리케이션 만들기
- 3. 서블릿, JSP, MVC 패턴 - JSP로 회원 관리 웹 애플리케이션 만들기
- 3. 서블릿, JSP, MVC 패턴 - MVC 패턴 - 개요
- 3. 서블릿, JSP, MVC 패턴 - MVC 패턴 - 적용
- 3. 서블릿, JSP, MVC 패턴 - MVC 패턴 - 한계

4. MVC 프레임워크 만들기

- 4. MVC 프레임워크 만들기 - 프론트 컨트롤러 패턴 소개
- 4. MVC 프레임워크 만들기 - 프론트 컨트롤러 도입 - v1
- 4. MVC 프레임워크 만들기 - View 분리 - v2
- 4. MVC 프레임워크 만들기 - Model 추가 - v3
- 4. MVC 프레임워크 만들기 - 단순하고 실용적인 컨트롤러 - v4
- 4. MVC 프레임워크 만들기 - 유연한 컨트롤러1 - v5
- 4. MVC 프레임워크 만들기 - 유연한 컨트롤러2 - v5

5. 스프링 MVC - 구조 이해

- 5. 스프링 MVC - 구조 이해 - 스프링 MVC 전체 구조
- 5. 스프링 MVC - 구조 이해 - 핸들러 매핑과 핸들러 어댑터
- 5. 스프링 MVC - 구조 이해 - 뷰 리졸버
- 5. 스프링 MVC - 구조 이해 - 스프링 MVC - 시작하기
- 5. 스프링 MVC - 구조 이해 - 스프링 MVC - 컨트롤러 통합
- 5. 스프링 MVC - 구조 이해 - 스프링 MVC - 실용적인 방식

6. 스프링 MVC - 기본 기능

- 6. 스프링 MVC - 기본 기능 - 프로젝트 생성
- 6. 스프링 MVC - 기본 기능 - 로깅 간단히 알아보기
- 6. 스프링 MVC - 기본 기능 - 요청 매핑
- 6. 스프링 MVC - 기본 기능 - 요청 매핑 - API 예시
- 6. 스프링 MVC - 기본 기능 - HTTP 요청 - 기본, 헤더 조회
- 6. 스프링 MVC - 기본 기능 - HTTP 요청 파라미터 - 쿼리 파라미터, HTML Form
- 6. 스프링 MVC - 기본 기능 - HTTP 요청 파라미터 - @RequestParam
- 6. 스프링 MVC - 기본 기능 - HTTP 요청 파라미터 - @ModelAttribute
- 6. 스프링 MVC - 기본 기능 - HTTP 요청 메시지 - 단순 텍스트
- 6. 스프링 MVC - 기본 기능 - HTTP 요청 메시지 - JSON

- 6. 스프링 MVC - 기본 기능 - HTTP 응답 - 정적 리소스, 뷰 템플릿
- 6. 스프링 MVC - 기본 기능 - HTTP 응답 - HTTP API, 메시지 바디에 직접 입력
- 6. 스프링 MVC - 기본 기능 - HTTP 메시지 컨버터
- 6. 스프링 MVC - 기본 기능 - 요청 매핑 핸들러 어댑터 구조

7. 스프링 MVC - 웹 페이지 만들기

- 7. 스프링 MVC - 웹 페이지 만들기 - 프로젝트 생성
- 7. 스프링 MVC - 웹 페이지 만들기 - 요구사항 분석
- 7. 스프링 MVC - 웹 페이지 만들기 - 상품 도메인 개발
- 7. 스프링 MVC - 웹 페이지 만들기 - 상품 서비스 HTML
- 7. 스프링 MVC - 웹 페이지 만들기 - 상품 목록 - 타임리프
- 7. 스프링 MVC - 웹 페이지 만들기 - 상품 상세
- 7. 스프링 MVC - 웹 페이지 만들기 - 상품 등록 폼
- 7. 스프링 MVC - 웹 페이지 만들기 - 상품 등록 처리 - @ModelAttribute
- 7. 스프링 MVC - 웹 페이지 만들기 - 상품 수정
- 7. 스프링 MVC - 웹 페이지 만들기 - PRG Post/Redirect/Get
- 7. 스프링 MVC - 웹 페이지 만들기 - RedirectAttributes

다음으로

스프링 MVC 2편 - 백엔드 웹 개발 활용 기술 안내

실제 예제에 단계적으로 기능을 발전시키며, 각 기능을 코드로 개발하면서 자연스럽게 학습

- 타임리프 뷰 템플릿 주요 기능 정리, 활용
 - 타임리프가 제공하는 기능 없이 사용할 때
 - 타임리프가 제공하는 기능을 학습하고 사용할 때 차이
 - 타임리프 뷰 템플릿 기능 정리, 다양한 기능 학습
- 메시지, 국제화 처리
 - 스프링의 메시지 처리 메커니즘 이해
 - 메시지, 국제화 처리 소개 및 구현
- 검증 - Validation
 - 컨트롤러에서 직접 검증 메커니즘 구현
 - 스프링이 제공하는 검증 메커니즘을 단계적으로 학습하고 적용
 - 수동 검증에서 애노테이션 기반의 검증까지

- 로그인 처리 - 쿠키, 세션
 - WAS가 제공하는 세션을 사용하지 않고, 직접 세션 구현해서 사용하기
 - WAS가 제공하는 세션 사용하기
- 로그인 처리 - 필터, 인터셉터
 - 개념과 차이 설명
 - 사용 예시
- 예외 처리
 - 예외 처리 매커니즘은 사실 매우 복잡해서 제대로 파악하기 쉽지 않다.
 - 제대로 파악하려면 서블릿의 예외처리 부터 스프링 부트가 제공하는 예외처리까지 복합적으로 이해해야 함
 - 순수한 서블릿 예외 처리 이해
 - 스프링 MVC가 제공하는 예외 처리 이해
 - 스프링 부트가 제공하는 예외 처리 이해
 - API 예외 처리
 - 예외 처리 완전 정복
- 자주 사용하는 기능, 기타
 - 나머지

여기까지 학습하면 스프링 **MVC** 완성!

실무 개발자들 보다 더 깊이있는 이해

로드맵 소개

스프링 완전 정복 시리즈(진행중)

스프링을 완전히 마스터 할 수 있는 로드맵

강의 목록

- 스프링 입문 - 코드로 배우는 스프링 부트, 웹 MVC, DB 접근 기술
- 스프링 핵심 원리 - 기본편
- 모든 개발자를 위한 HTTP 웹 기본 지식
- 스프링 웹 MVC 1편
- 스프링 웹 MVC 2편(5월)
- 스프링 DB 접근 기술 (예정)
- 실전! 스프링 부트 (예정)

실전 REST API 설계, 개발 강의 요청이 많아서, 고민중

스프링 부트와 JPA 실무 완전 정복 로드맵

최신 실무 기술로 웹 애플리케이션을 만들어보면서 학습하고 싶으면 **스프링 부트와 JPA 실무 완전 정복 로드맵**을 추천

URL: <https://www.infllearn.com/roadmaps/149>

강의 목록

- 자바 ORM 표준 JPA 프로그래밍
- 실전! 스프링 부트와 JPA 활용1 - 웹 애플리케이션 개발
- 실전! 스프링 부트와 JPA 활용2 - API 개발과 성능 최적화
- 실전! 스프링 데이터 JPA
- 실전! Querydsl

Q: 전체 코스 순서는 어떻게 듣는게 좋나요?

목표: 최신 기술의 자바 백엔드 개발자

- **스프링 완전 정복 로드맵**
 - 스프링 입문 - 코드로 배우는 스프링 부트, 웹 MVC, DB 접근 기술
 - 스프링 핵심 원리 - 기본편
 - 모든 개발자를 위한 HTTP 웹 기본 지식
 - 스프링 웹 MVC 1편
 - 스프링 웹 MVC 2편
- **스프링 부트와 JPA 실무 완전 정복 로드맵**
- **학자형 코스 vs 야생형 코스**
 - 학자형 코스: 이론을 먼저 차근차근 쌓아올린 다음에 실무 활용으로 넘어가는 방식
 - 야생형 코스: 일단 실무에서 어떤 기술을 어떤식으로 활용하는지, 깊이가 부족해도 일단 경험해본 다음에 필요에 의해서 이론 기술들을 공부하는 방식

추천 학습 방법

스프링 입문과 스프링 핵심 원리를 듣고 나면 스프링으로 개발하는 가장 중요한 기본 지식을 쌓은 상태가 됩니다. 그래서 바로 **스프링 부트와 JPA 실무 완전 정복 로드맵**에 들어가는 것도 좋은 선택입니다.

스프링을 실무에서 어떤 식으로 활용해서 개발하는지 먼저 배워두고, 이후에 **스프링 완전정복 로드맵**을 통해 스프링 MVC나 스프링 데이터 접근 기술 같은 부분은 더 깊이있게 학습하시면 됩니다.

문제 해결 방안과 야생형

야생형 코스가 맞는 사람도 있고, 학자형 코스가 맞는 사람도 있음