*077. Front Controller 도입

요청 →
BoardAddController
← 응답(redirect)

요청 →
BoardDetailController
← 응답(redirect)
↕
detail.jsp

요청 →
BoardDetailController
← 응답(redirect)
↕(오류발생)
detail.jsp

*077. Front Controller 도입 후 : Architecture

>Controller의 공통기능을 처리, Client와의 결합을 줄인다. ("Low coupling"
refresh, redirect,jsp 객체, 오류처리등

client
- refresh
- redirect
- jst 실행결과
- 오류 내용

요청 →
← 응답(redirect)

Front Controller

JSP
- 결과 페이지 생성
- 오류 내용 출력

위임 → BoardAddController
위임 → BoardDetailController
위임 → BoardUpdateController
↓
Controller의 공통 기능을 Front Controller에 맡김으로써 기능 구현이 간결해진다.

*Facade 디자인 패턴을 적용한 예이다.
(Front)
객체간의 복잡한 작업 과정을 캡슐화함으로써 코딩이 간결해진다.
사용객체와의 결합을 캡슐으로써 유지보수가 쉬워진다. "Low coupling"

예) 종합개설 종합 생성 담당자 Facade 패턴 적용 결과 1개 → 종합 생성 담당자
고객 실명인증 담당자 → 고객 → 고객담당자 → 실명인증 담당자
카드개설 담당자 → 카드개설 담당자

*77 Front Controller 도입 후 : Arhcitecture

client
- refresh
- redirect
- jst 실행결과
- 오류 내용

요청 → /service/*
← 응답(redirect)

<<front Controller>>
DispatcherServlet
↕
JSP
- 결과 페이지 생성
- 오류 내용 출력

위임 → /board/add → BoardAddController
위임 → /board/detail → BoardDetailController
위임 → /board/update → BoardUpdateController

"Page Controller"

http://localhost:8888/app

MyServlet

1.애노테이션을 붙이기 서블릿 소스코드가 있다면
@WebServlet

2.DD파일에 설정(Web.xml)
<servlet></servlet>
<servlet-mapping>...</servlet-mapping>

3.자바코드로 직접 서블릿 객체 등록 서블릿 소스코드가 없다면
Myservlet servlet = new Myservlet();
Dynamic config = sc.addServlet(servlet);
config.addMapping("/hello");

*078.페이지 컨트롤러를 POGO로 만들기

client
- refresh
- redirect
- jst 실행결과
- 오류 내용

요청 → /service/* 서블릿
← 응답(redirect)

<<front Controller>>
DispatcherServlet
↕
JSP
- 결과 페이지 생성
- 오류 내용 출력

execute( ) →
← viewName

<<interface>>
Controller
△
| ---------- <concrete>>
BoardListController
|
| ---------- BoardAddController
|

프론트 컨트롤러에서 요청을 받기 때문에
페이지 컨트롤러는 굳이 서블릿이 될 이유가 없다.