

BoostCourse

(Full-Stack Developer)

메인화면 동작

- TODO, DOING, DONE 3가지 상태로 구분.
- 세가지 상태는 동일한 크기와 동일한 간격으로 배치.
- 각 카드UI간의 간격은 위/아래모두 동일한 margin으로 유지.
- 새로운 카드(할일)가 추가되면 아래에 동일한 카드가 추가됨.
- 카드의 갯수는 계속 증가할 수 있음.
- 각 카드에는
할일제목/등록날짜/담당자/우선순위를 표현.
- 먼저 등록한 카드(할일)가 위에 노출된다.
나중에 등록한 카드는 아래 추가되어 노출된다.
- 각 카드에서 오른쪽 화살표를 누르면
오른쪽 상태로 이동한다. 예를들어
TODO에서 각 카드의 화살표를 누르면
Doing으로 이동한다. done상태로가면
오른쪽 화살표가 없어진다.
(버튼을 눌렀을때 카드가 이동하는 건
화면의 새로고침 없이 자바스크립트만으로
구현되어야 한다.)

나의 해야할 일들

새로운 TODO 등록

TODO	DOING	DONE
자바스크립트 공부하기 등록날짜: 2018.03.10, 홍길동, 우선순위 1 →	다음달 계획서 작성방법 고민하기 등록날짜: 2018.01.22, 홍길동, 우선순위 1 →	CSS학습 등록날짜: 2018.01.10, 홍길동, 우선순위 1
리포트 제출하기 등록날짜: 2018.02.20, 홍길동, 우선순위 1 →	데모페이지 개발 등록날짜: 2018.01.10, 홍길동, 우선순위 1 →	JAVA 객체지향 설계 복습 등록날짜: 2018.01.11, 홍길동, 우선순위 1
	웹 관련 책 찾아보기 등록날짜: 2018.02.20, 홍길동, 우선순위 1 →	이사할 곳 알아보기 등록날짜: 2018.01.03, 홍길동, 우선순위 1
	웹폰트 공부하기 등록날짜: 2018.01.20, 홍길동, 우선순위 1 →	

할일 등록

- 할일제목, 담당자, 우선순위 를 입력해야 함.
- 할일제목은 24자까지 입력해야 함.
- 할일제목과 담당자는 text 형태로 입력받음.
- 우선순위는 라디오 박스로 한가지만 선택가능.
- 제출을 누르면 서버로 데이터가 전송된다.
전송된 이후에는 메인페이지에 추가된 내용이 포함되어 노출됨.
- 할일 등록시 필요한 데이터는 '할일제목', '담당자', '우선순위', '등록날짜' 이다. 이중 등록날짜는 서버에서 자동으로 현재 날짜를 DB에 저장한다.
- 내용지우기(reset)를 누르면 모든 내용이 다시 초기화됨.
- 이전을 누르면 메인페이지로 화면이 전환됨.
- 3가지 항목은 모두 필수로 입력되어야 한다.

할일 등록

어떤일인가요?

swift 공부하기(24자까지)

누가 할일인가요?

홍길동

우선순위를 선택하세요

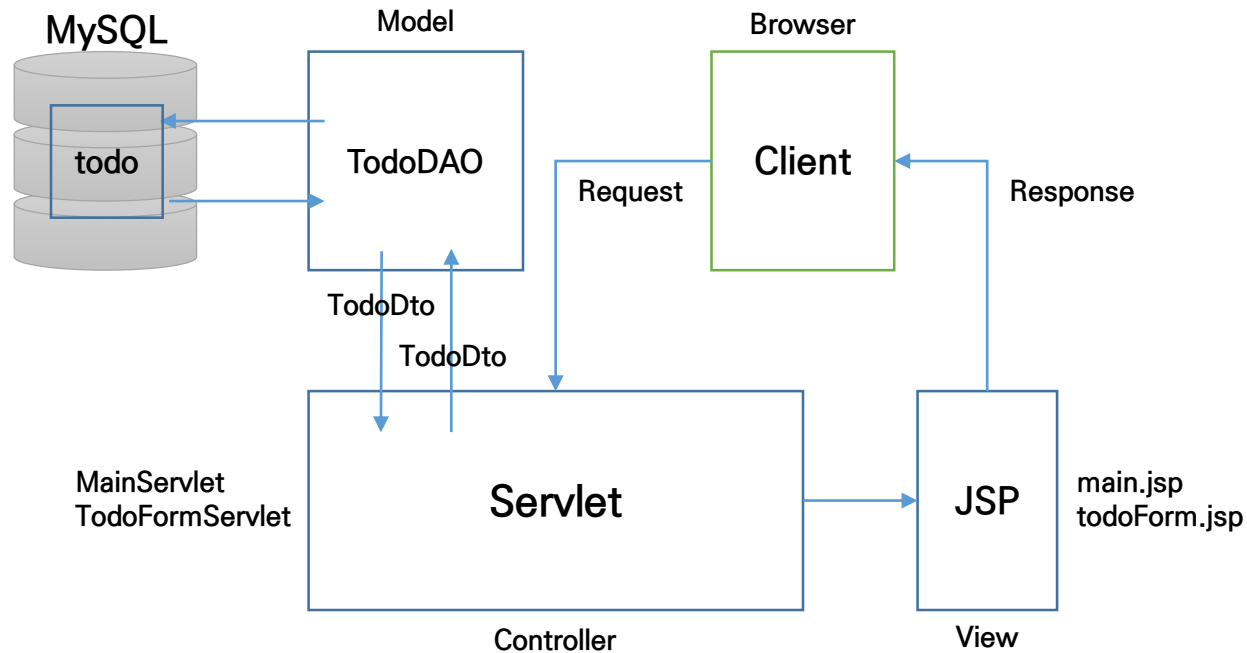
☐ 1순위 ☐ 2순위 ☐ 3순위

< 이전

제출

내용지우기

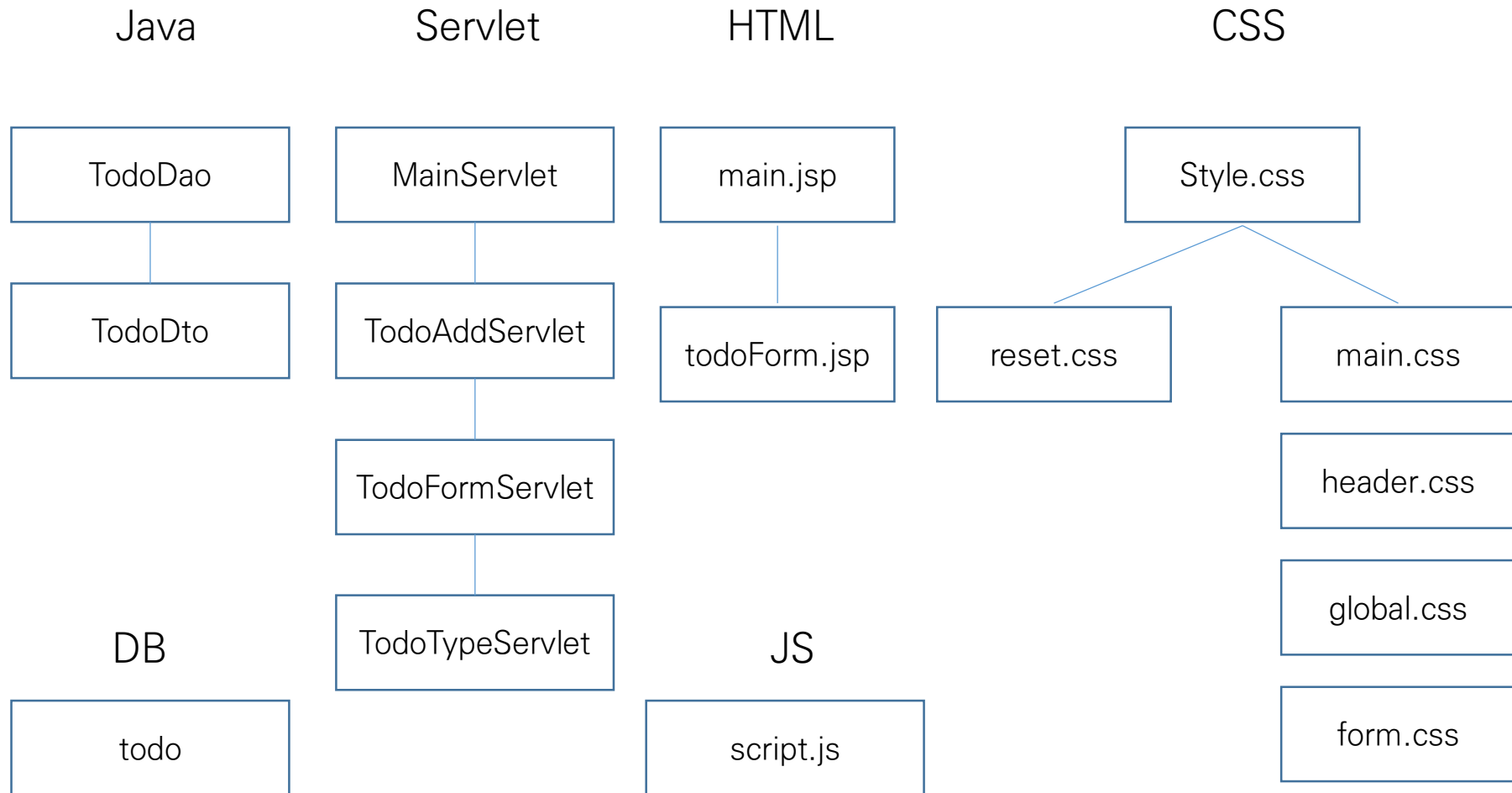
시스템 구조



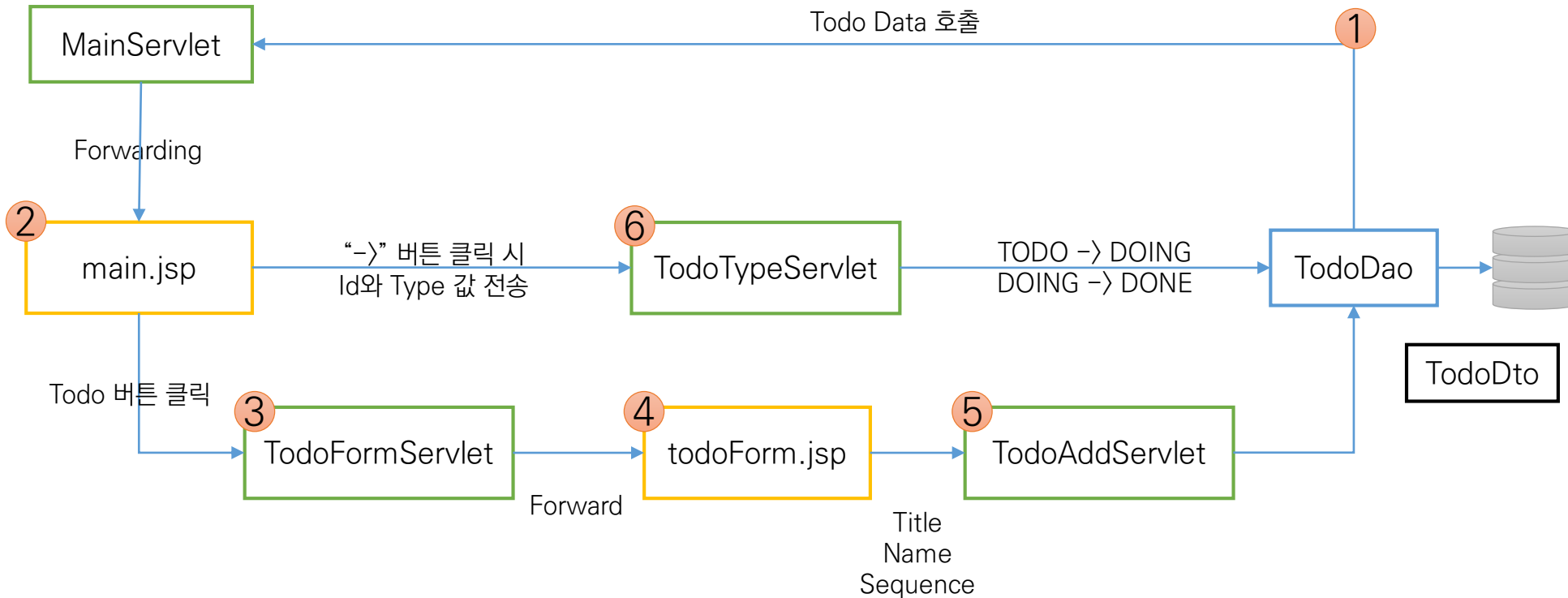
DataFlow

- main.jsp에서 결과를 조회하여 보여준다.
- 새로운 todo등록 버튼을 클릭하면 TodoFormServlet이 실행, TodoFormServlet은 todoForm.jsp로 포워딩하여 할 일 등록 화면을 보여준다.
- 할 일 등록 폼에서 값을 입력, 제출 버튼 클릭 시 post방식으로 TodoAddServlet으로 값이 전달, TodoAddServlet에서는 TodoDao를 이용하여 테이블에 저장, 메인 화면으로 리다이렉트한다.
- 메인화면에서 todo 상태변경버튼(-))을 클릭하면 TodoTypeServlet에게 id와 상태 값을 전달하여 다음 상태로 (현재 상태가 Todo라면 Doing으로 Doing이라면 Done으로) TodoDao를 이용해서 변경한다.

파일 구조



서비스 흐름도



[1] MainServlet

- 프로그램 실행 시 TodoDao를 통해 데이터 호출
- MainServlet에서 main.jsp로 forwarding

[2] main.jsp

- Data의 type에 따라 데이터 출력
- Todo 버튼 클릭 시 TodoFormServlet로 호출
- todo 상태버튼 (->) 클릭 시 TodoTypeServlet에게 Todo의 id와 상태 값을 전달

[3] TodoFormServlet

- todoForm.jsp로 포워딩

[4] todoForm.jsp

- 할일 등록 폼에서 값을 입력
- 제출 버튼을 누르면 post방식으로 TodoAddServlet으로 값 전달

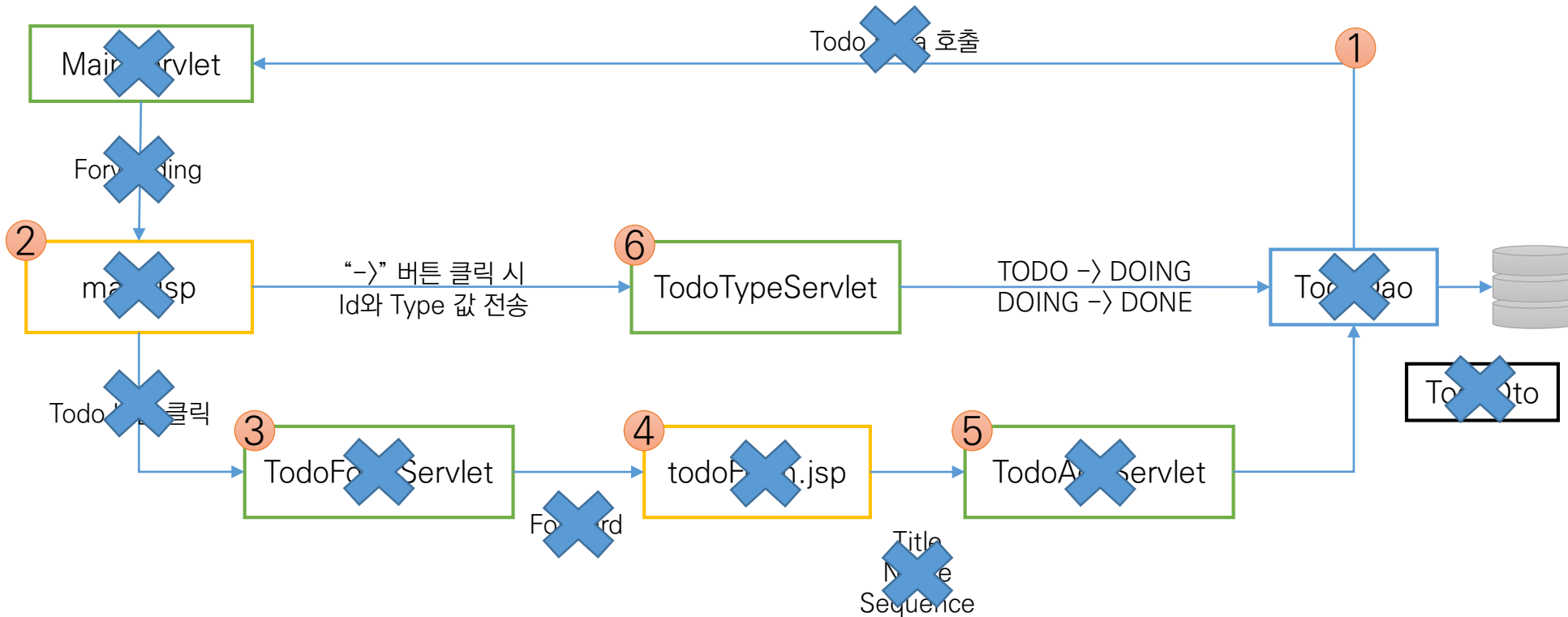
[5] TodoAddServlet

- TodoDao를 이용해서 테이블에 저장하고 메인화면으로 리다이렉트

[6] TodoTypeServlet

- Todo의 id의 상태 값을 전달, 다음 상태로 (TODO라면 DOING, DOING이라면 DONE로) TodoDao를 이용해서 변경

서비스 흐름도



[1] MainServlet

- 프로그램 실행 시 TodoDao를 통해 데이터 호출
- MainServlet에서 main.jsp로 forwarding

[2] main.jsp

- Data의 type에 따라 데이터 출력
- Todo 버튼 클릭 시 TodoFormServlet로 호출
- todo 상태버튼 (->) 클릭 시 TodoTypeServlet에게 Todo의 id와 상태 값을 전달

[3] TodoFormServlet

- todoForm.jsp로 포워딩

[4] todoForm.jsp

- 할일 등록 폼에서 값을 입력
- 제출 버튼을 누르면 post방식으로 TodoAddServlet으로 값 전달

[5] TodoAddServlet

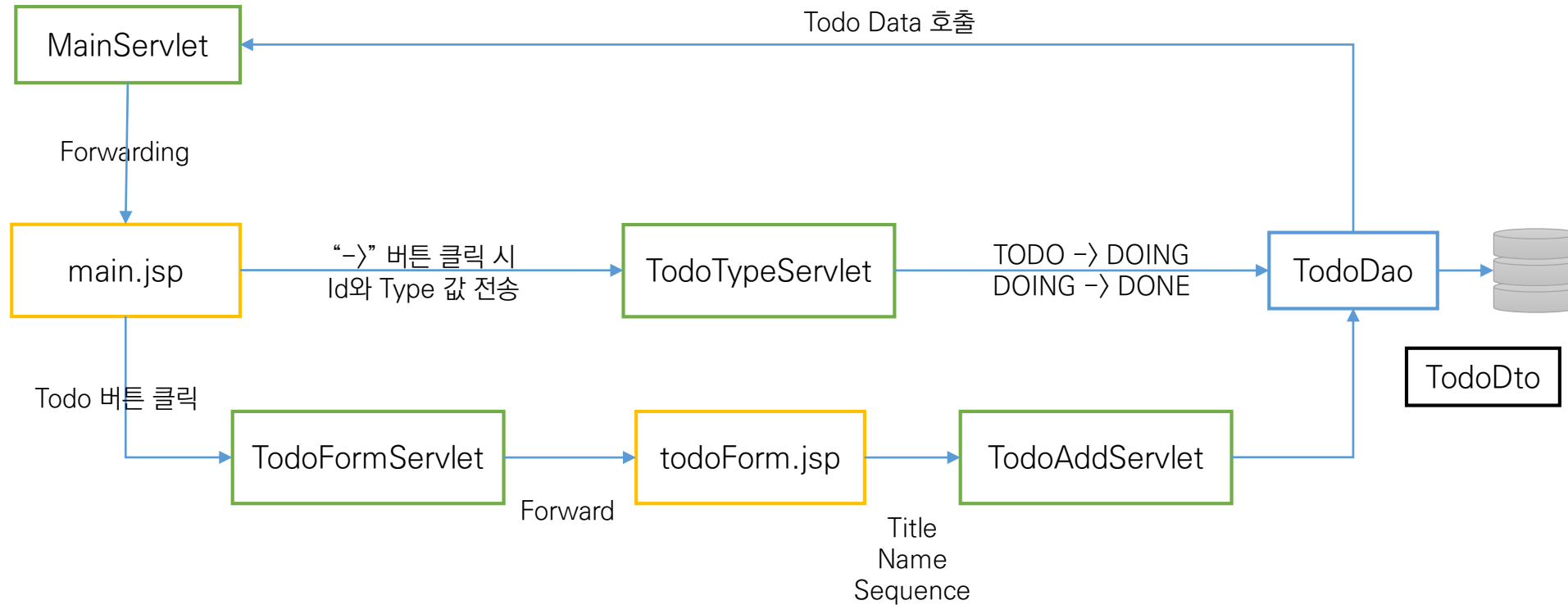
- TodoDao를 이용해서 테이블에 저장하고 메인화면으로 리다이렉트

[6] TodoTypeServlet

- Todo의 id의 상태 값을 전달, 다음 상태로 (TODO라면 DOING, DOING이라면 DONE로) TodoDao를 이용해서 변경

[1] main.jsp

- addEventListener 이벤트 등록
- DOM API를 사용
(appendChild 또는 insertAdjacentHTML 등을 활용)
- Ajax를 통해 서버에 변경상태를 보내고 그 결과를 받은 이후에 DOM 변경작업
- 버튼을 눌렀을 때 카드가 이동하는 건 화면의 새로고침 없이 자바 스크립트로 구현

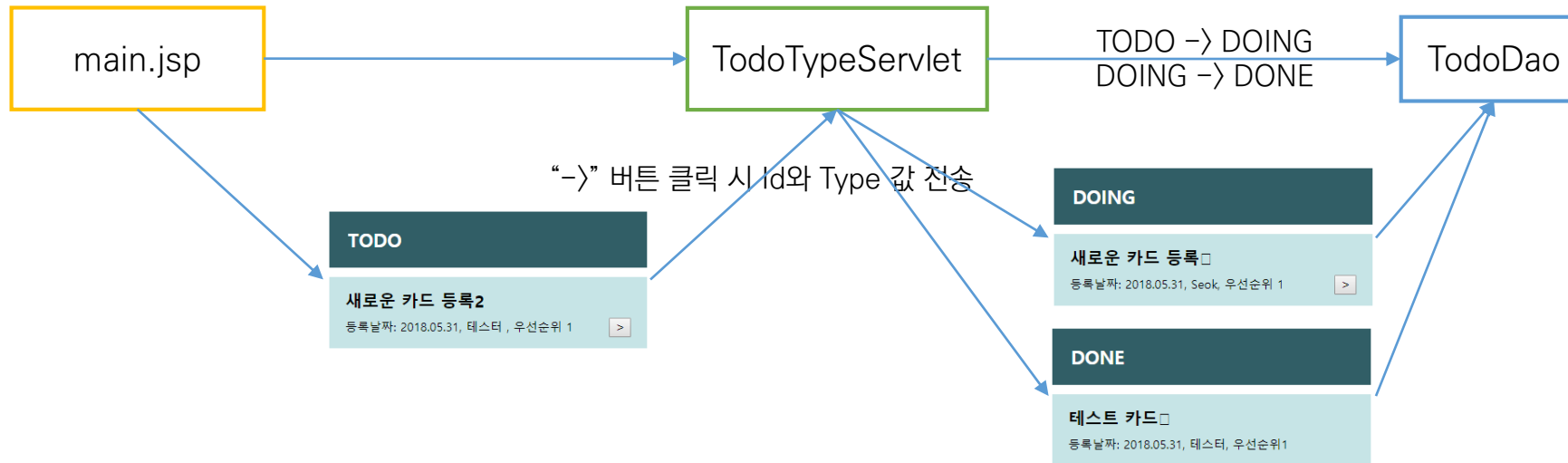


[1] initPage

- 카드의 버튼에 이벤트 등록

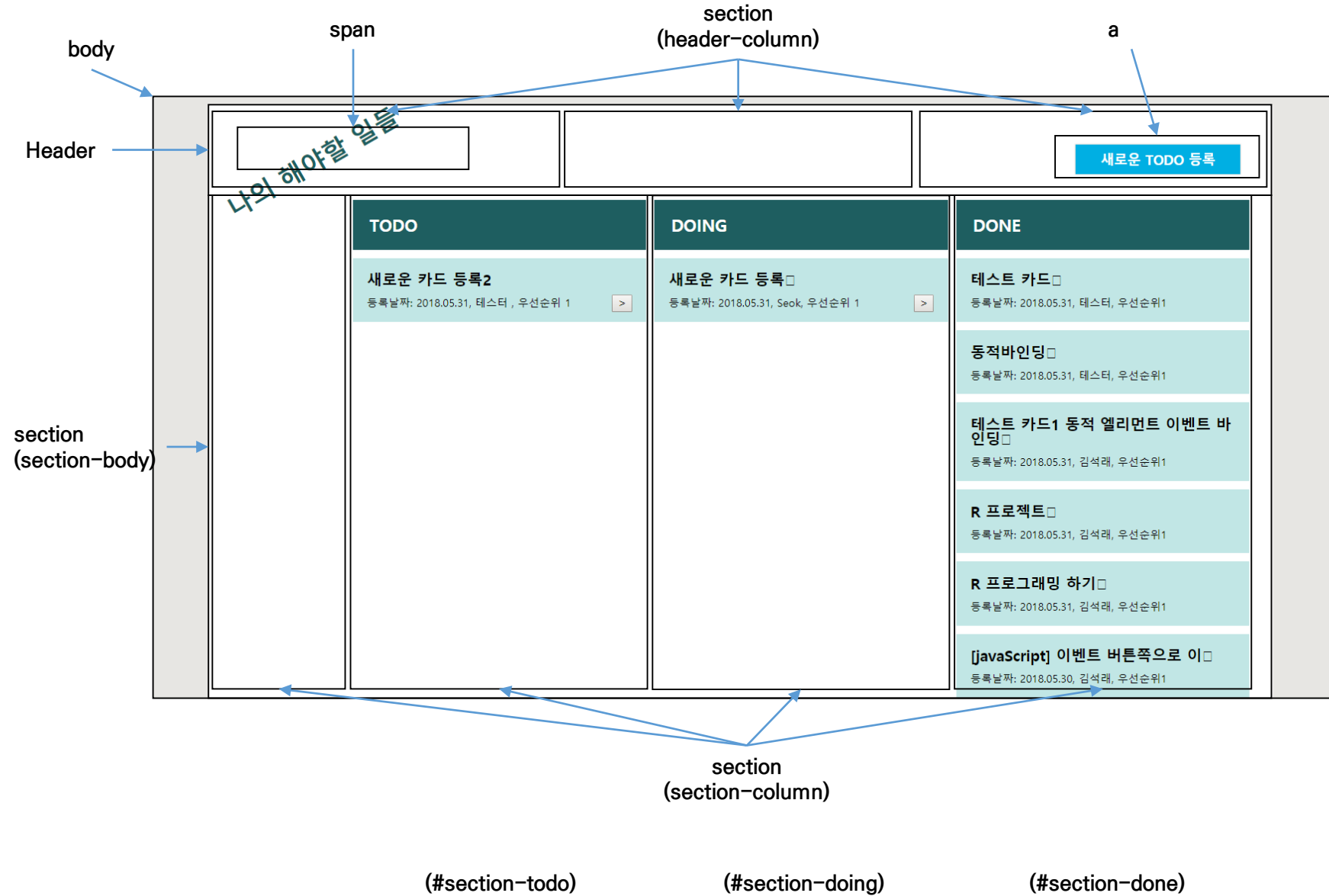
[2] typeChange

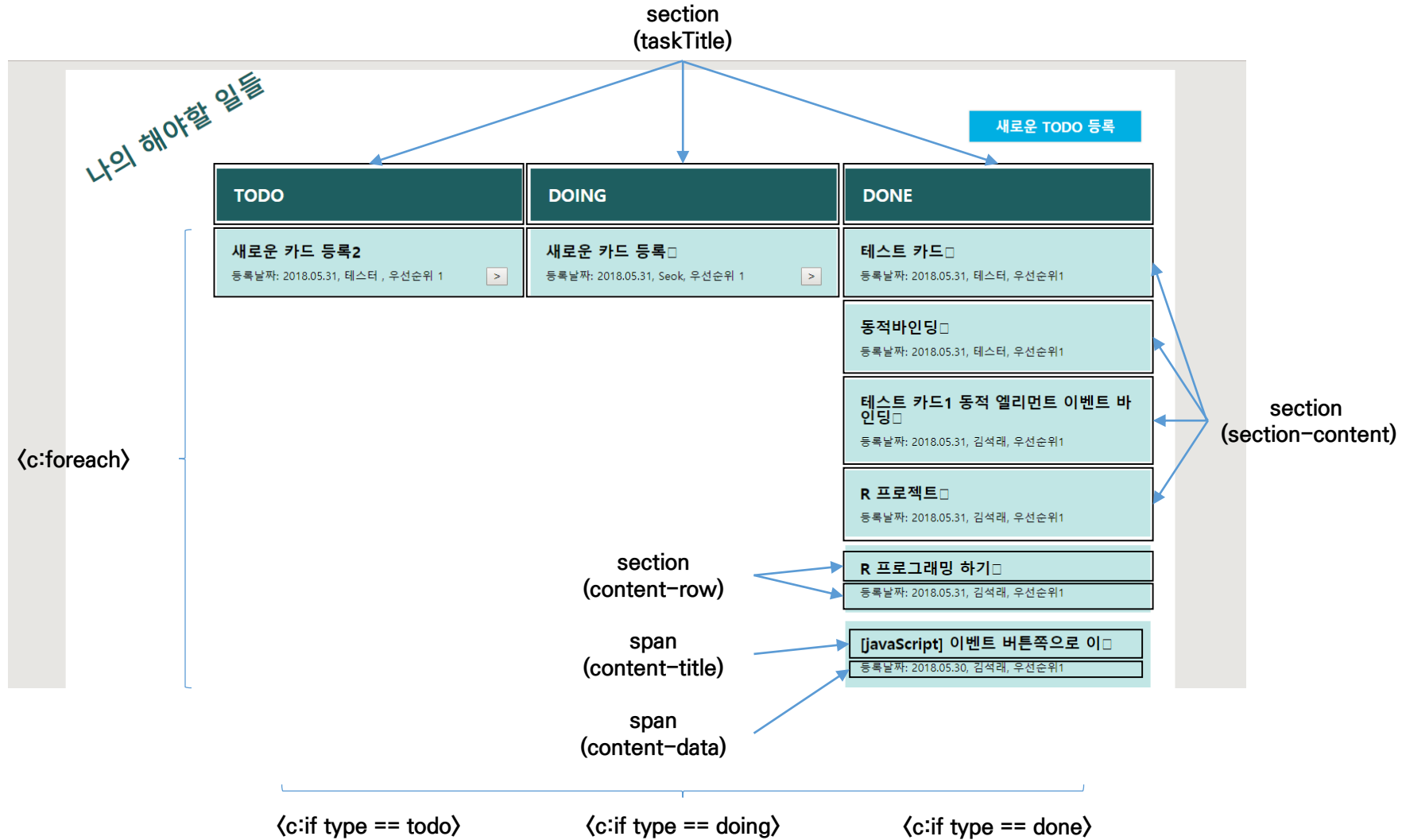
- 비동기 통신 (XMLHttpRequest)
- 서버로 Type 변경 데이터 송신
- 카드 이동, 카드 삭제



ISSUE:

Todo 카드의 버튼을 눌렀을 때
현 Type Column에서
다음 Type Column으로 이동
(Ajax를 이용한 비동기)



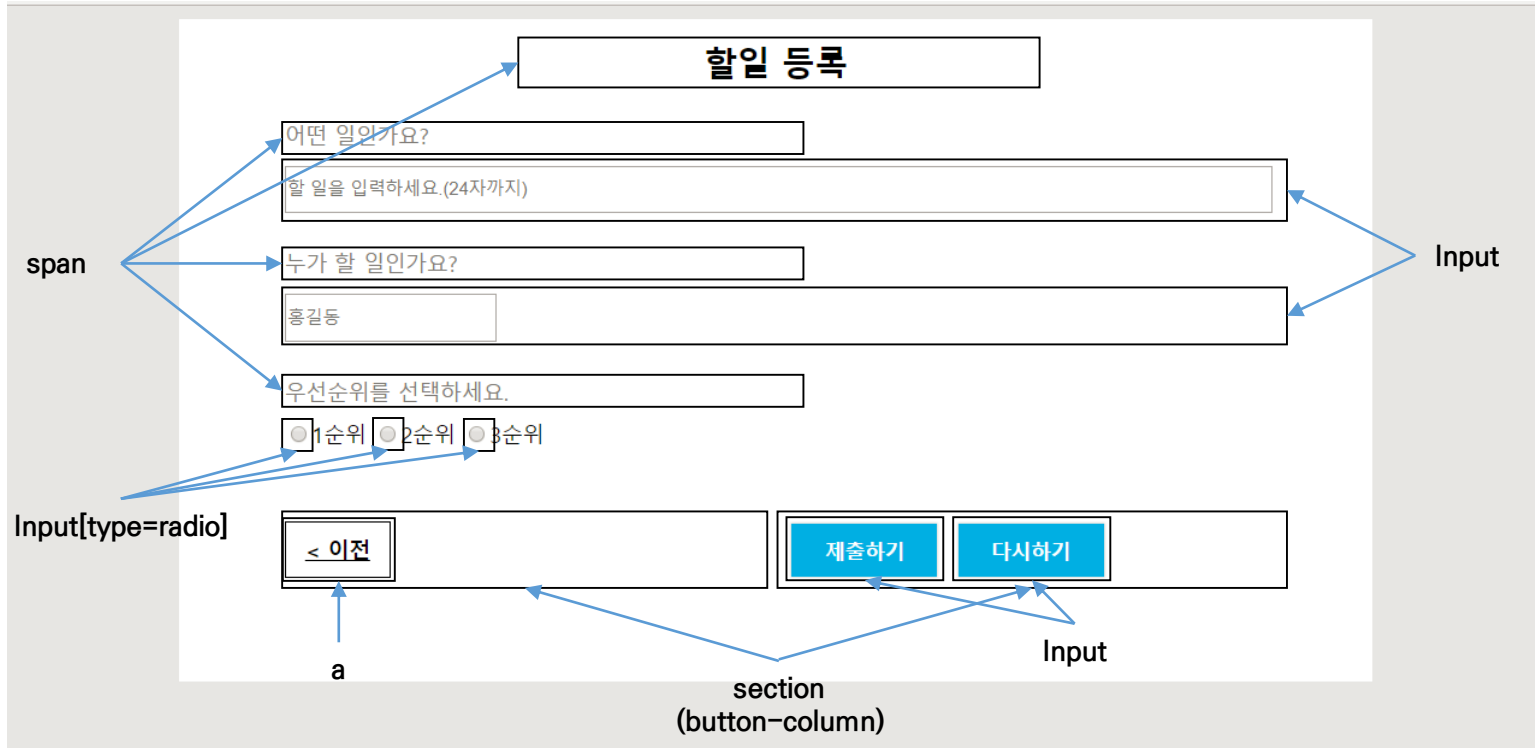


The diagram illustrates the layout of the `todoForm.jsp` page. It features a main container labeled `body` which contains a central form area. This form area is composed of several sections:

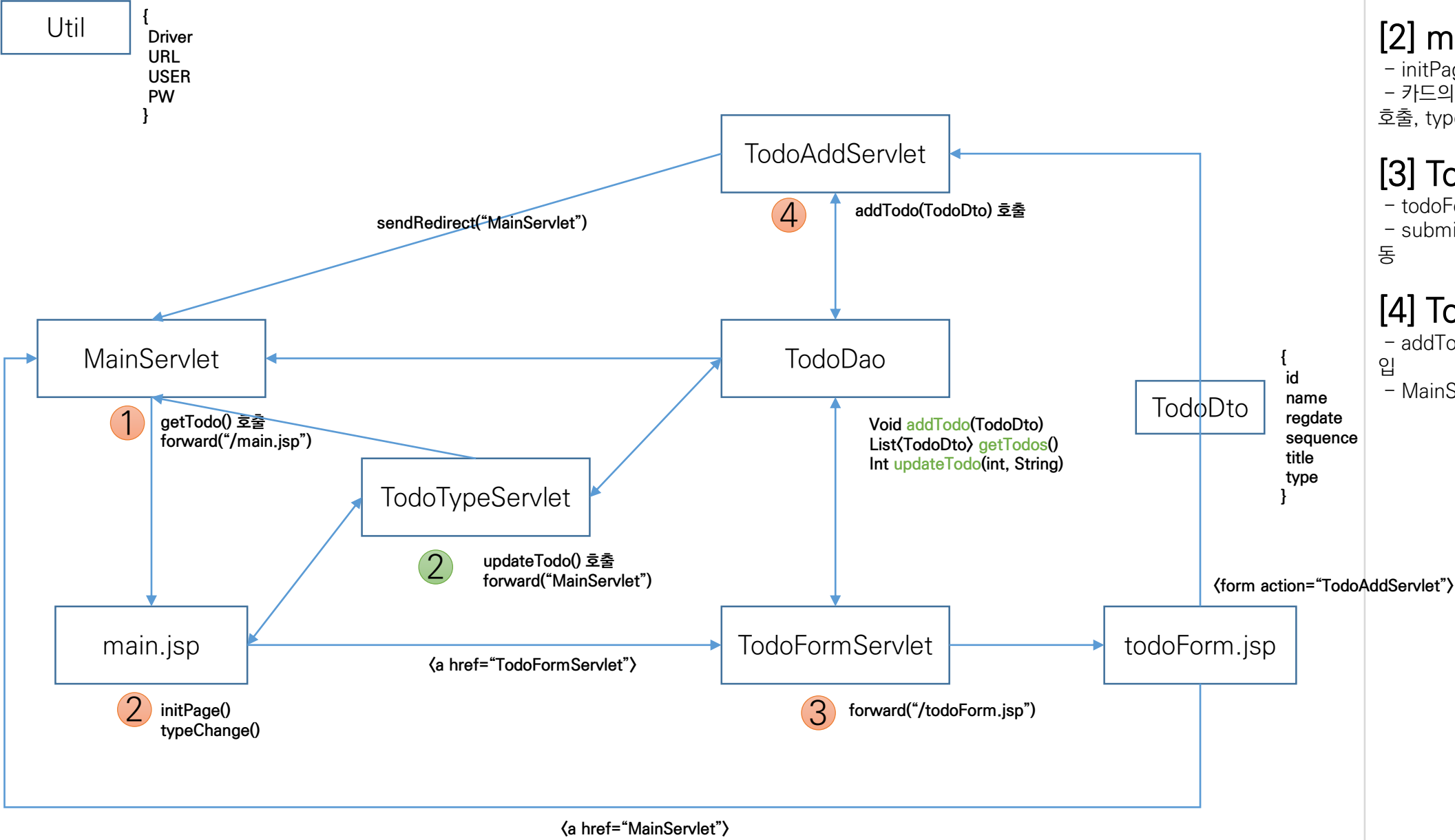
- A header section titled **할일 등록** (Task Registration).
- A section for "어떤 일인가요?" (What task is it?) containing a text input field with the placeholder "할 일을 입력하세요.(24자까지)".
- A section for "누가 할 일인가요?" (Who will do the task?) containing a text input field with the placeholder "홍길동".
- A section for "우선순위를 선택하세요." (Select priority) with three radio button options: ☐ 1순위, ☐ 2순위, and ☐ 3순위.
- A footer section containing a button labeled `< 이전` (Previous) and two buttons labeled `제출하기` (Submit) and `다시하기` (Reset).

Annotations on the right side of the diagram identify the components:

- `body`: Points to the outermost container.
- `Section (section-form)`: Points to the header section.
- `Section (form-column)`: Points to the input fields for task description and assignee.
- `Section (form-column)`: Points to the priority selection section.
- `Section (form-column)`: Points to the footer section with navigation buttons.



서블릿 서비스 흐름



[1] MainServlet

- getTodo() 호출로 데이터 호출
- main.jsp로 forward() 호출

[2] main.jsp

- initPage() 호출로 데이터 출력
- 카드의 버튼 클릭으로 typeChange() 호출, type 변경 및 카드 이동

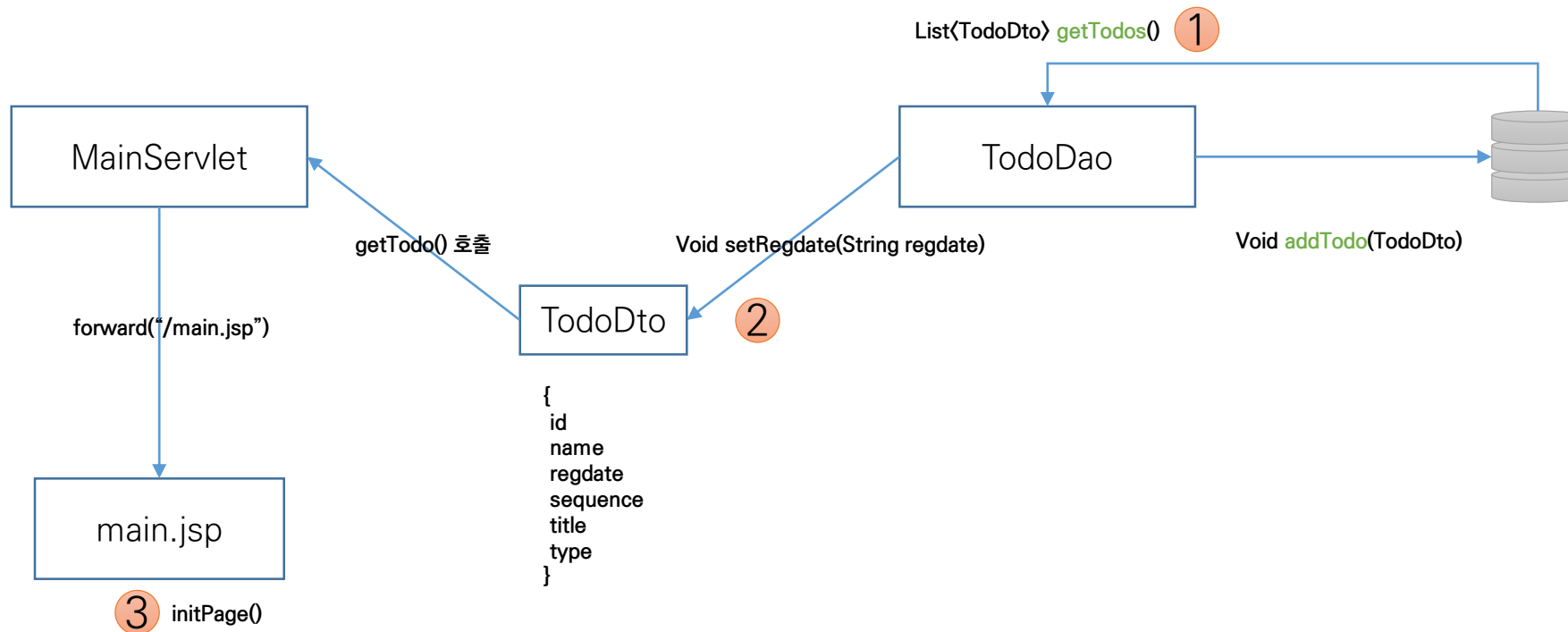
[3] TodoFormServlet

- todoForm.jsp 로 forward() 호출
- submit 버튼으로 TodoAddServlet 이동

[4] TodoAddServlet

- addTodo(TodoDto) 호출로 데이터 삽입
- MainServlet으로 redirect()

날짜 데이터 출력을 어떻게 할 것인가?



[1] TodoDao

- getTodos() 함수를 통해 데이터베이스에서 데이터 호출

[2] TodoDto

- setRegdate(String regdate) 함수를 통해 regdate값 저장

- MySQL의 DATETIME 타입의 포맷 "YYYY-MM-DD HH:MM:SS"

- 문자열 split()함수를 통해 앞 부분만 저장 "YYYY-MM-DD"

- 문자열 replace()함수를 통해 "YYYY.MM.DD"로 결과 값 저장

[3] MainServlet

- getTodo()함수를 통해 호출한 데이터를 main.jsp 페이지에 출력