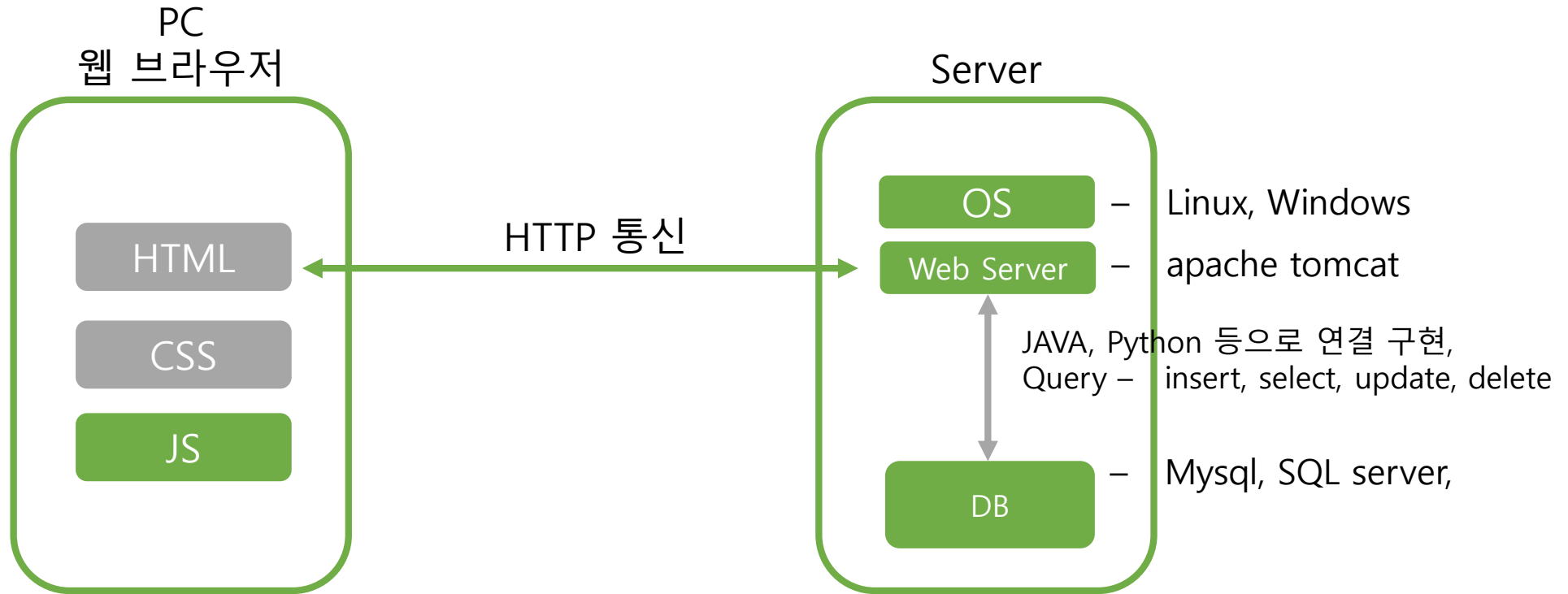


인턴 교육 일지1 (1/2) - 정인정

- 웹서비스 구조와 스프링 프레임워크

1 웹 서비스 구조

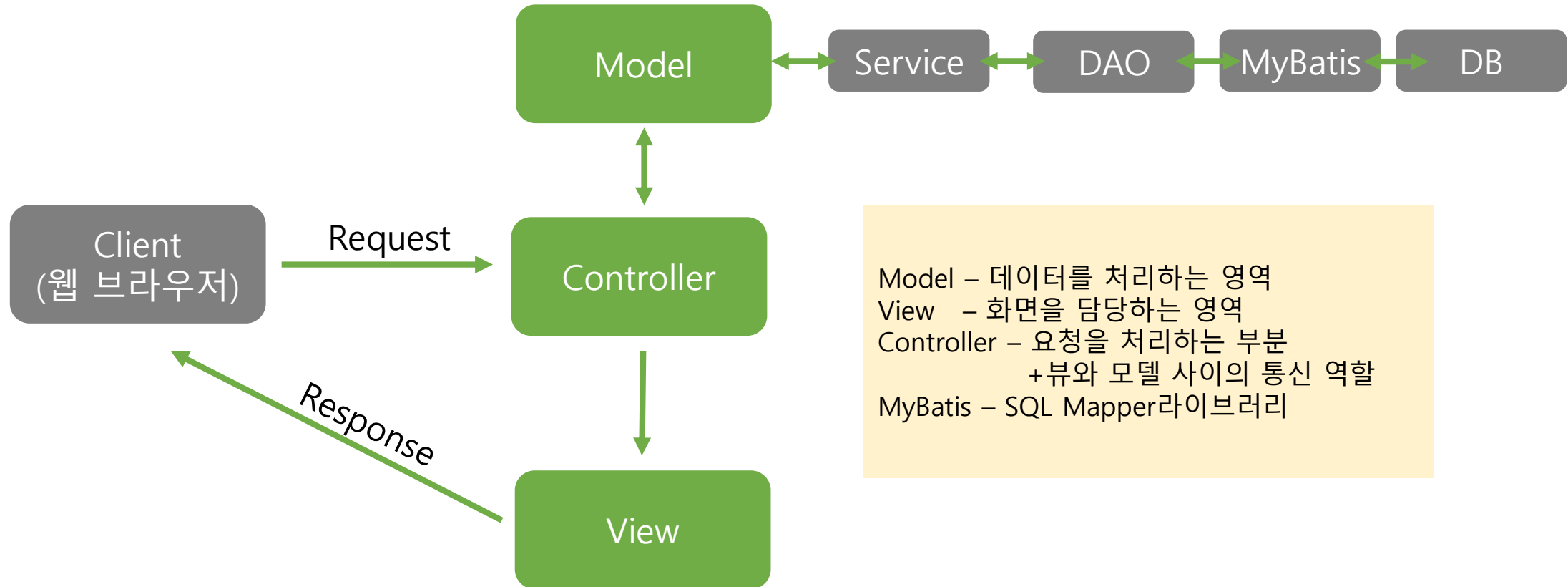
PC와 Server



2 Spring Framework

스프링 구조 - MVC

- 각각의 모듈은 역할이 명확해진다.
- 유지보수가 쉬워지고 개발자간의 협업이 원활해진다.



Model - 데이터를 처리하는 영역
View - 화면을 담당하는 영역
Controller - 요청을 처리하는 부분
+ 뷰와 모델 사이의 통신 역할
MyBatis - SQL Mapper 라이브러리

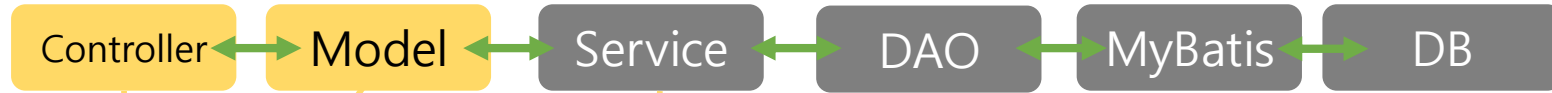
3 게시판 구현하기

DB – 테이블 생성

```
CREATE TABLE TB_BOARD (  
    IDX NUMBER PRIMARY KEY,  
    PARENT_IDX NUMBER,  
    TITLE VARCHAR2(100) NOT NULL,  
    CONTENTS VARCHAR2(4000) NOT NULL,  
    HIT_CNT NUMBER NOT NULL,  
    DEL_GB VARCHAR2(1) DEFAULT 'N' NOT NULL,  
    CREA_DTM DATE DEFAULT SYSDATE NOT NULL,  
    CREA_ID VARCHAR2(30) NOT NULL );
```

3 게시판 구현하기

Controller 구현



```
@Controller public class SampleController {
    Logger log = Logger.getLogger(this.getClass());
    @Resource(name="sampleService")
    private SampleService sampleService;
    @RequestMapping(value="/sample/openSampleBoardList.do")
    public ModelAndView openSampleBoardList(Map<string,object> commandMap, Model model) throws Exception{
        List<Map<String, Object>> list = sampleService.selectBoardList(commandMap);
        model.addAttribute("list", list);
        return "openSampleBoardList";
    }
}
```

3 게시판 구현하기

Service 구현

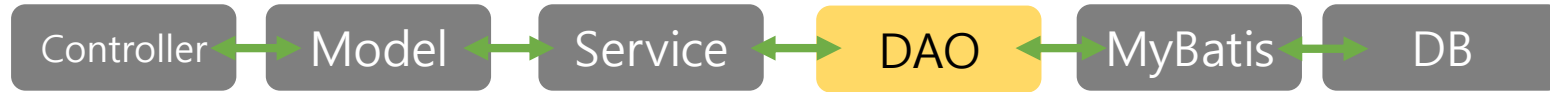


```
public interface SampleService { }

@Service("sampleService")
public class SampleServiceImpl implements SampleService{
    Logger log = Logger.getLogger(this.getClass());
    @Resource(name="sampleDAO")
    private SampleDAO sampleDAO;
    @Override public List<Map<String, Object>> selectBoardList(Map<String, Object> map) throws Exception {
        return sampleDAO.selectBoardList(map);
    }
}
```

3 게시판 구현하기

DAO 구현

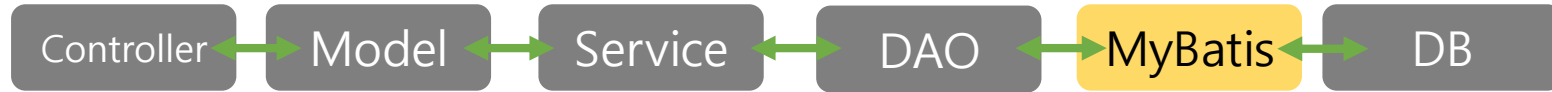


```
@Repository("sampleDAO")
public class SampleDAO extends AbstractDAO{ }

@Repository("sampleDAO") public class SampleDAO extends AbstractDAO{
    @SuppressWarnings("unchecked")
    public List<Map<String, Object>> selectBoardList(Map<String, Object> map) throws Exception{
        return (List<Map<String, Object>>)selectList("sample.selectBoardList", map);
    }
}
```

3 게시판 구현하기

SQL mapper.xml 구현



```
<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
  <mapper namespace="sample">
    <select id="selectBoardList" parameterType="hashmap" resultType="hashmap">
      <![CDATA[ SELECT IDX, TITLE, HIT_CNT, CREA_DTM FROM TB_BOARD WHERE DEL_GB = 'N' ORDER BY IDX DESC ]]>
    </select>
  </mapper>
```


3 게시판 구현하기

openSampleBoardList.jsp 구현 C:forEach를 통해 controller로 부터 넘겨받은 list객체를 뷰에 나타낼 수 있음.

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html> <html> <head>
    <title>first</title>
    <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %> <%@ taglib prefix="fn" uri="http://java.
sun.com/jsp/jstl/functions" %> </head>
    <body>
        <table style="border:1px solid #ccc"> <thead>
            <tr>
                <th scope="col">글번호</th><th scope="col">제목</th>
                <th scope="col">조회수</th><th scope="col">작성일</th>
            </tr> </thead>
            <tbody>
                <c:choose>
                    <c:when test="${fn:length(list) > 0}">
                        <c:forEach items="${list}" var="row">
                            <tr>
                                <td>${row.IDX }</td> <td>${row.TITLE }</td>
                                <td>${row.HIT_CNT }</td> <td>${row.CREA_DTM }</td>
                            </tr>
                        </c:forEach></c:when> <c:otherwise>
                            <tr> <td colspan="4">조회된 결과가 없습니다.</td>
                        </tr> </c:otherwise> </c:choose> </tbody></table></body></html>
```

+ References

1

2

- <https://gmlwjd9405.github.io/2018/10/26/spring-framework.html>
- <https://doublesprogramming.tistory.com/174>
- <https://cloudstudying.kr/lectures/235>

3

- <https://addio3305.tistory.com/72?category=772645>