

Spring Board 2

Google Mail

1. pom.xml

The screenshot shows the Maven Repository website. The top navigation bar includes the MVN REPOSITORY logo, a search bar, and links for Categories, Popular, and Contact Us. On the left, there's a sidebar with 'Indexed Artifacts (32.9M)' and a line graph showing project growth from 2006 to 2018. The main content area displays the breadcrumb 'Home » org.springframework » spring-context-support » 5.3.19'. Below this is the 'Spring Context Support » 5.3.19' header with the Spring logo. The description states: 'Support classes for integrating common third-party libraries into a Spring application context'.

The screenshot shows the Maven Repository website for the JavaMail API JAR. The breadcrumb is 'Home » javax.mail » javax.mail-api » 1.6.2'. The header is 'JavaMail API JAR » 1.6.2' with the Java logo. The description is 'JavaMail API JAR'. Below the description is a table with the following data:

License	CDDL GPL 2.0
Categories	Mail Clients Java Specifications
Tags	standard javax mail client api specs

The screenshot shows the Maven Repository website for the JavaMail API. The breadcrumb is 'Home » com.sun.mail » javax.mail » 1.6.2'. The header is 'JavaMail API » 1.6.2' with the Sun/Oracle logo. The description is 'JavaMail API'. Below the description is a table with the following data:

License	CDDL GPL 2.0
Categories	Mail Clients
Tags	mail client
Date	Aug 20, 2018

```

<!-- Mail -->
  <!-- <https://mvnrepository.com/artifact/org.springframework/spring-context-support> -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context-support</artifactId>
    <version>5.3.19</version>
  </dependency>
  <!-- <https://mvnrepository.com/artifact/javax.mail/javax.mail-api> -->
  <dependency>
    <groupId>javax.mail</groupId>
    <artifactId>javax.mail-api</artifactId>
    <version>1.6.2</version>
  </dependency>
  <!-- <https://mvnrepository.com/artifact/com.sun.mail/javax.mail> -->
  <dependency>
    <groupId>com.sun.mail</groupId>
    <artifactId>javax.mail</artifactId>
    <version>1.6.2</version>
  </dependency>

```

2. web.xml

```

    and Filters -->
    <!-- /WEB-INF/spring/*.xml -->
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/spring/root-context.xml
                  /WEB-INF/spring/security-context.xml
                  /WEB-INF/spring/mail-context.xml
    </param-value>
  </context-param>

```

3. mail-context.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           https://www.springframework.org/schema/beans/spring-beans.xsd">

  <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">
    <property name="host" value="smtp.gmail.com"></property>
    <property name="port" value="587"></property>
    <property name="username" value="yoonkeunsoo@gmail.com"></property>
    <property name="password" value="본인 앱키"></property>
    <property name="javaMailProperties">
      <props>
        <prop key="mail.smtp.auth">true</prop>
        <prop key="mail.smtp.starttls.enable">true</prop>
        <prop key="mail.transport.protocol">smtp</prop>
        <prop key="mail.debug">true</prop>
        <prop key="mail.smtp.ssl.trust">smtp.gmail.com</prop>
        <prop key="mail.smtp.ssl.protocols">TLSv1.2</prop>
      </props>
    </property>
  </bean>

</beans>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           https://www.springframework.org/schema/beans/spring-beans.xsd">

  <bean id="mailSender" class="org.springframework.mail.javamail.JavaMailSenderImpl">
    <property name="host" value="smtp.gmail.com"></property>
    <property name="port" value="587"></property>
    <property name="username" value="yoonkeunsoo@gmail.com"></property>
    <property name="password" value="본인 앱"></property>
    <property name="javaMailProperties">
      <props>
        <prop key="mail.smtp.auth">true</prop>
        <prop key="mail.smtp.starttls.enable">true</prop>
        <prop key="mail.transport.protocol">smtp</prop>
        <prop key="mail.debug">true</prop>
        <prop key="mail.smtp.ssl.trust">smtp.gmail.com</prop>
        <prop key="mail.smtp.ssl.protocols">TLSv1.2</prop>
      </props>
    </property>
  </bean>

</beans>

```

4. Service

@Async 스프링이 제공하는 비동기식 메소드 지원 애너테이션이다.

```

package com.carshop.mail;

public interface MailService {

    void sendMail(String to, String subject, String body);
}

package com.carshop.mail;

import javax.mail.internet.MimeMessage;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.mail.javamail.JavaMailSender;
import org.springframework.mail.javamail.MimeMessageHelper;
import org.springframework.scheduling.annotation.Async;
import org.springframework.stereotype.Service;

@Service
public class GoogleMailServiceImpl implements MailService {

    @Autowired
    private JavaMailSender mailSender;

    @Async
    public void sendMail(String to, String subject, String body) {
        MimeMessage message = mailSender.createMimeMessage();

        try {
            MimeMessageHelper messageHelper = new MimeMessageHelper(message, true, "UTF-8");
            messageHelper.setFrom("yoonkeunsoo@gmail.com", "고객센터");
            messageHelper.setSubject(subject);
            messageHelper.setTo(to);
            messageHelper.setText(body);
            mailSender.send(message);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

5. Controller

```

package com.carshop.mail;

import java.io.PrintWriter;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.scheduling.annotation.EnableAsync;

```

```

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

@RequestMapping("mail")
@EnableAsync
@Controller
public class MailController {

    @Autowired
    private MailService mailService;

    @RequestMapping(value = "/sendmail", method = RequestMethod.GET)
    public void sendMail(HttpServletRequest request, HttpServletResponse response) throws Exception {
        request.setCharacterEncoding("utf-8");
        response.setContentType("text/html; charset=utf-8");
        PrintWriter out = response.getWriter();
        mailService.sendMail("austiny@snu.ac.kr", "테스트 메일입니다.", "내용을 없습니다.");
        out.println("메일 전송 완료");
    }

    // 폼 메일 전송 연습

    @RequestMapping(value = "/form", method=RequestMethod.GET)
    public String mailform() {
        return "mail/form";
    }

    @RequestMapping(value = "/form", method=RequestMethod.POST)
    public void mailform(@RequestParam Map<String, Object> map) {

        String to = (String) map.get("to");
        String subject = (String) map.get("subject");
        String body = (String) map.get("body");

        mailService.sendMail(to, subject, body);

    }

}

```

6. View

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

```

```

<h1> 메일 전송 폼 </h1>

<form action="/mail/form" method="POST">
  <p>제목 : <input type="text" name="subject" />
  <p>내용 : <input type="text" name="body" />
  <p>받는이 : <input type="text" name="to" />
  <input type="submit" value="메일 전송" />
  <input type="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>
</form>

</body>
</html>

```

이메일 보내기

```

@Autowired
private MailService mailService;

```

```

@PostMapping("/join")
public String submitForm(@ModelAttribute("NewUser") User user) {

    //스프링은 반드시 password를 암호화 하여 저장해야만 로그인을 할수 있는게 기본이다.

    String encodedPassword = bcryptPasswordEncoder.encode(user.getPassword());
    user.setPassword(encodedPassword);

    userService.setNewUser(user);

    String to = user.getEmail();
    String subject = user.getUsername() + " 님 회원 가입을 환영합니다.";
    String body = "저희 사이트에 회원으로 가입하셔서 감사드립니다. ";

    mailService.sendMail(to, subject, body);

    return "redirect:/login";
}

```

게시판

하단에 답변달기 처리 ajax

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@ taglib prefix="sec"
    uri="http://www.springframework.org/security/tags"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>게시물 보기</title>
<script src="https://code.jquery.com/jquery-3.6.3.min.js"
    integrity="sha256-pvPw+upLPUjgMXy0G+800xUf+/Im1MZjXxxgOcBQBXU="
    crossorigin="anonymous"></script>

<script>
    function replyNewFunction() {
        $.ajax({
            type : "POST",
            url : "/boards/replynew",
            data : {
                bid : document.getElementById('bid').value,
                bwriter : document.getElementById('bwriter').value,
                bcontent : document.getElementById('bcontent').value
            },
            beforeSend : function(xhr) { /*데이터를 전송하기 전에 헤더에 csrf값을 설정한다*/
                xhr.setRequestHeader("${_csrf.headerName}", "${_csrf.token}");
            },
            success : function(result) {
                alert("답변 등록 성공");
            },
            error : function(request, status, error) {
                alert(request.status + " " + request.responseText);
            }
        });
        window.location.reload();
    }
</script>

</head>
<body>

    <!-- 주 게시물 -->
    <div class="card">
        <h5 class="card-header">${board.btitle}</h5>
        <div class="card-body">
            <h5 class="card-title">${board.bcontent}</h5>
            <br>
            <br>
            <br>
            <div class="d-flex justify-content-end">
                <p class="btn btn-success btn-sm">조회수 [ ${board.bview} ]</p>
                <p class="btn btn-primary btn-sm">좋아요 [ ${board.blake} ]</p>
                <p class="btn btn-danger btn-sm">싫어요 [ ${board.bhate} ]</p>
            </div>
        </div>
    </div>

```

```

        <div class="d-flex justify-content-end badge bg-light text-dark">${board.bdate}
    </div>
    </div>
</div>

<!-- 답변 게시물 -->

<div class="card">
    <div class="card-body">답변 게시물 리스트</div>

    <!-- 답변 등록 -->
    <sec:authentication property="principal" var="user" />
    <input id="bwriter" type="hidden" value="${user.username}"/>
    <input id="bid" type="hidden" name="bid" value="${board.bid}">

    <div class="card text-center">
        <div class="card-header">
            작성자 : ${user.username}
        </div>
        <div class="card-body">
            <h5 class="card-title">답변 등록</h5>
            <p class="card-text">
                <textarea name="bcontent" id="bcontent" rows="3" class="form-control"></textarea>
            </p>
            <a href="#" class="btn btn-primary" onclick="replyNewFunction()">답변등록</a>
        </div>
        <div class="card-footer text-muted">
            2 days ago
        </div>
    </div>

</div>

<div class="my-5"></div>

</body>
</html>

```

BoardController 에 매서드 작성

```

@ResponseBody
@RequestMapping("/replynew")
public void replynew(@RequestParam Map<String, Object> map) {
    boardService.replynewBoard(map);
}

```

Service


```

package com.carshop.boards;

import java.util.List;
import java.util.Map;

public interface BoardService {

    void setNewBoard(Board board);

    List<Board> getAllBoardList();

    Board getBoardById(String bid);

    void replynewBoard(Map map);
}

public void replynewBoard(Map map) {
    boardRepository.replynewBoard(map);
}

```

repository

```

public void replynewBoard(Map map) {
    this.sqlSessionTemplate.insert("board.insert_reply", map);
}

package com.carshop.boards;

import java.util.List;
import java.util.Map;

public interface BoardRepository {

    void setNewBoard(Board board);

    List<Board> getAllBoardList();

    Board getBoardById(String bid);

    void replynewBoard(Map map);

}

```

xml

```

<insert id="insert_reply" parameterType="hashMap"
    useGeneratedKeys="true" keyProperty="bid">
    <![CDATA[
        INSERT INTO board
        (bpid, bcontent, bwriter)
        values
        ({bid}, #{bcontent}, #{bwriter})
    ]]>
</insert>

```

boardServiceImpl.java

```

package com.carshop.boards;

import java.util.List;
import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class BoardServiceImpl implements BoardService{

    @Autowired
    BoardRepository boardRepository;

    public void setNewBoard(Board board) {
        boardRepository.setNewBoard(board);
    }

    public List<Board> getAllBoardList() {
        return boardRepository.getAllBoardList();
    }

    public Board getBoardById(String bid) {

        return boardRepository.getBoardById(bid);
    }

    public void replynewBoard(Map map) {
        boardRepository.replynewBoard(map);
    }

    public List<Board> getReplyById(String bid) {
        return boardRepository.getReplyById(bid);
    }
}

```

board.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@ taglib prefix="sec"
    uri="http://www.springframework.org/security/tags"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>게시물 보기</title>
<script src="https://code.jquery.com/jquery-3.6.3.min.js"
    integrity="sha256-pvPw+upLPUjgMXy0G+800xUf+/Im1MZjXxxgOcBQBXU="
    crossorigin="anonymous"></script>

<script>
    function replyNewFunction() {
        $.ajax({
            type : "POST",
            url : "/boards/replynew",
            data : {
                bid : document.getElementById('bid').value,
                bwriter : document.getElementById('bwriter').value,
                bcontent : document.getElementById('bcontent').value
            },
            beforeSend : function(xhr) { /*데이터를 전송하기 전에 헤더에 csrf값을 설정한다*/
                xhr.setRequestHeader("${_csrf.headerName}", "${_csrf.token}");
            },
            success : function(result) {
                alert("답변 등록 성공");
            },
            error : function(request, status, error) {
                alert(request.status + " " + request.responseText);
            }
        });
        window.location.reload();
    }
</script>

</head>
<body>

    <!-- 주 게시물 -->
    <div class="card">
        <h5 class="card-header">${board.btitle}</h5>
        <div class="card-body">
            <h5 class="card-title">${board.bcontent}</h5>
            <br>
            <br>
            <br>
            <div class="d-flex justify-content-end">
                <p class="btn btn-success btn-sm">조회수 [ ${board.bview} ]</p>
                <p class="btn btn-primary btn-sm">좋아요 [ ${board.blake} ]</p>
                <p class="btn btn-danger btn-sm">싫어요 [ ${board.bhate} ]</p>
            </div>
        </div>
    </div>

```

```

        <div class="d-flex justify-content-end badge bg-light text-dark">${board.bdate}
    </div>
    </div>
</div>

<!-- 답변 게시물 -->

<div class="card">
    <div class="card-body">답변 게시물 리스트</div>
    <b>${cnt}개의 답변이 있습니다.</b>
    <c:forEach items="${replyList}" var="reply">

        ${reply.bcontent}
        ${reply.bdate}
        ${reply.bwriter}

    </c:forEach>
    <!-- 답변 등록 -->
    <sec:authentication property="principal" var="user" />
    <input id="bwriter" type="hidden" value="${user.username}"/>
    <input id="bid" type="hidden" name="bid" value="${board.bid}">

    <div class="card text-center">
        <div class="card-header">
            작성자 : ${user.username}
        </div>
        <div class="card-body">
            <h5 class="card-title">답변 등록</h5>
            <p class="card-text">
                <textarea name="bcontent" id="bcontent" rows="3" class="form-control"></textarea>
            </p>
            <a href="#" class="btn btn-primary" onclick="replyNewFunction()">답변등록</a>
        </div>
        <div class="card-footer text-muted">
            2 days ago
        </div>
    </div>

</div>

<div class="my-5"></div>

</body>
</html>

```

```

<select id="select_reply" parameterType="String" resultType="com.carshop.boards.Board">
    <![CDATA[
        SELECT *
        FROM board
        WHERE bpid = #{bid}
        ORDER BY bid DESC
    ]]>

```

```
    ]]>
</select>

public List<Board> getReplyById(String bid) {
    return this.sqlSessionTemplate.selectList("board.select_reply", bid);
}
```