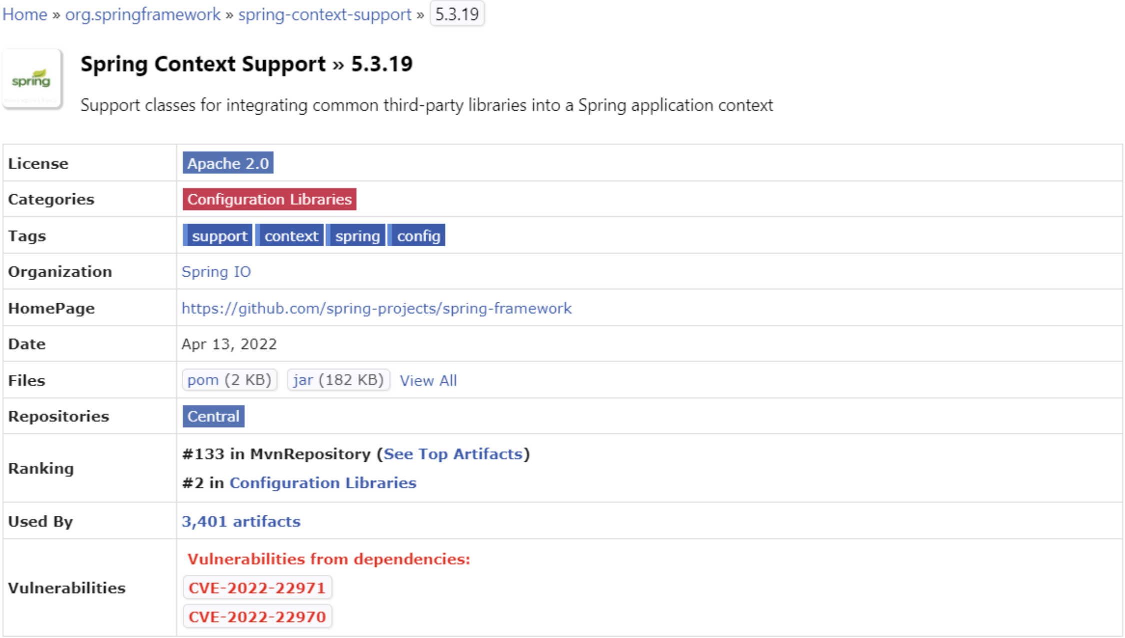
Spring Day 18

Spring

▼ Google Mail

▼ 환경설정

1. **pom.xml 설정.**

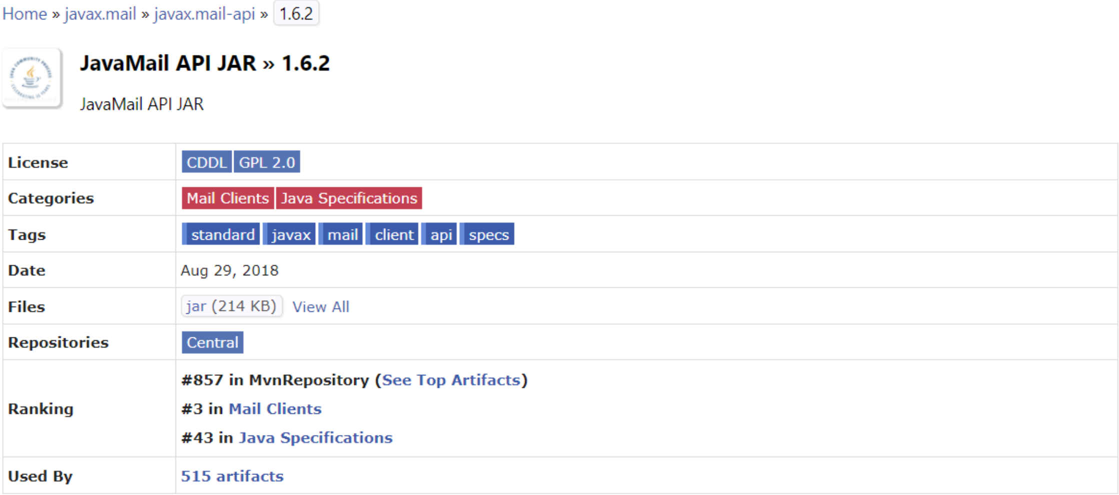


Home » org.springframework » spring-context-support » 5.3.19

Spring Context Support » 5.3.19

Support classes for integrating common third-party libraries into a Spring application context

License	Apache 2.0
Categories	Configuration Libraries
Tags	support context spring config
Organization	Spring IO
HomePage	https://github.com/spring-projects/spring-framework
Date	Apr 13, 2022
Files	pom (2 KB) jar (182 KB) View All
Repositories	Central
Ranking	**#133 in MvnRepository (See Top Artifacts)** **#2 in Configuration Libraries**
Used By	**3,401 artifacts**
Vulnerabilities	**Vulnerabilities from dependencies:** **CVE-2022-22971** **CVE-2022-22970**

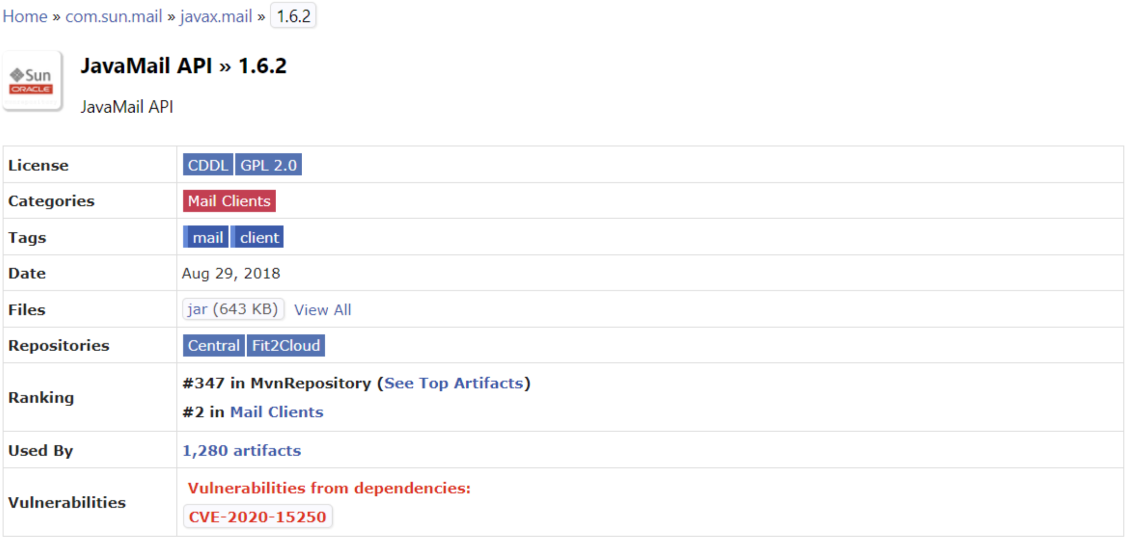


Home » javax.mail » javax.mail-api » 1.6.2

JavaMail API JAR » 1.6.2

JavaMail API JAR

License	CDDL GPL 2.0
Categories	Mail Clients Java Specifications
Tags	standard javax mail client api specs
Date	Aug 29, 2018
Files	jar (214 KB) View All
Repositories	Central
Ranking	**#857 in MvnRepository (See Top Artifacts)** **#3 in Mail Clients** **#43 in Java Specifications**
Used By	**515 artifacts**



Home » com.sun.mail » javax.mail » 1.6.2

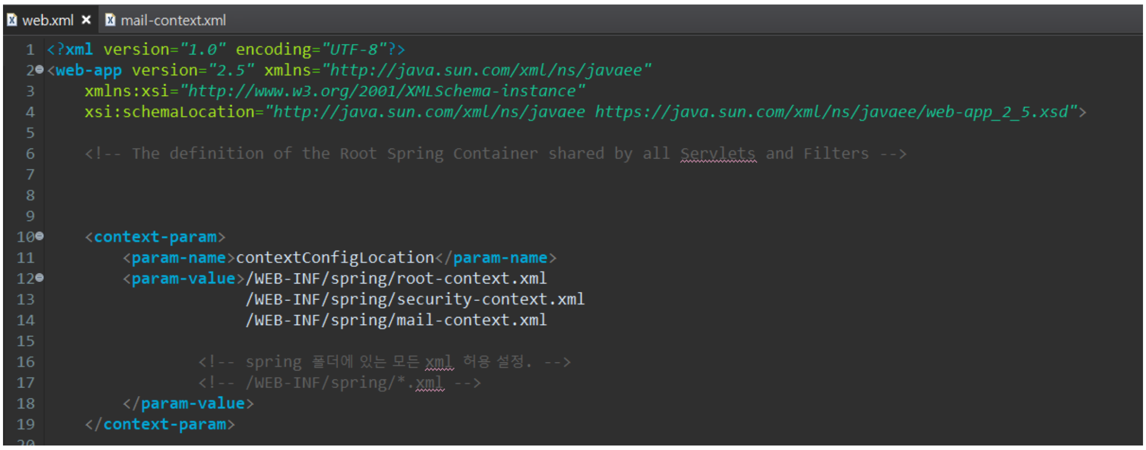
JavaMail API » 1.6.2

JavaMail API

License	CDDL GPL 2.0
Categories	Mail Clients
Tags	mail client
Date	Aug 29, 2018
Files	jar (643 KB) View All
Repositories	Central Fit2Cloud
Ranking	**#347 in MvnRepository (See Top Artifacts)** **#2 in Mail Clients**
Used By	**1,280 artifacts**
Vulnerabilities	**Vulnerabilities from dependencies:** **CVE-2020-15250**

```
<!-- 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



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee https://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

    <!-- The definition of the Root Spring Container shared by all Servlets and Filters -->



    <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

                <!-- spring 폴더에 있는 모든 xml 허용 설정. -->
                <!-- /WEB-INF/spring/*.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 name="port" value="587"/>
  <property name="username" value="difbfl4750@gmail.com"/>
  <property name="password" value="앱 비밀번호"/>
  <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.protocol">TLSv1.2</prop>
    </props>
  </property>
</bean>

</beans>
```

MailService.java

```
package com.carshop.mail;

public interface MailService {

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

}
```

GoogleMailServiceImpl.java

```
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 //비동기 메소드 작성 어노테이션
  @Override
  public void sendMail(String to, String subject, String body) {
    MimeMessage message = mailSender.createMimeMessage();

    try {
      MimeMessageHelper messageHelper = new MimeMessageHelper(message, true, "UTF-8");
      messageHelper.setFrom("difbfl4750@gmail.com","고객센터");
      messageHelper.setSubject(subject);
      messageHelper.setTo(to);
      messageHelper.setText(body);
      mailSender.send(message);
    } catch (Exception e) {
      e.printStackTrace();
    }
  }

}
```

MailController.java

```
package com.carshop.mail;

import java.io.IOException;
import java.io.PrintWriter;
import java.util.Map;

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.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;


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

  @Autowired
  private MailService mailService;

  @GetMapping("/send")
  public void sendMail(HttpServletRequest request, HttpServletResponse response) throws IOException {
    request.setCharacterEncoding("utf-8");
    response.setContentType("text/html;charset=utf-8");
    PrintWriter out = response.getWriter();
    mailService.sendMail("difbfl4750@gmail.com", "테스트 메일입니다.", "내용");
    out.println("메일 전송 완료");
  }

  // 폼 메일 전송 연습
  @GetMapping("/form")
  public String mailform() {
    return "mail/form";
  }

  @PostMapping("/form")
  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);
  }



}
```

form.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>메일 전송 폼</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="hidden" name="${_csrf.parameterName}" value="${_csrf.token}"/>
<input type="submit" value="메일 전송">
</form>


</body>
</html>
```

▼ 회원가입시 이메일 전송처리

UsersController.java

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

    //Spring은 기본적으로 password를 암호화하여 DB에 등록해야만 로그인을 할 수 있도록 설정되어 있다.

    String encodedPassword = bCryptPasswordEncoder.encode(user.getPassword());
    user.setPassword(encodedPassword);
    userService.setNewUser(user);

    String to = user.getUemail();
    String subject = user.getUname() + "님 환영합니다[CarShop]";
    String body = "회원가입이 완료되었습니다.";

    mailService.sendMail(to, subject, body);

    return "login";
  }
```

▼ 게시판 등록시 이메일 전송처리

BoardController.java

```
@PostMapping("/addBoard")
  public String submitAddBoardForm(@ModelAttribute("NewBoard") Board board) {

    boardService.setNewBoard(board);

    User user = userService.existUsername(board.getBwriter());

    String to = user.getUemail();
    String subject = board.getBwriter() + "님의 게시물이 등록되었습니다.";
    String body = board.getBcontent();

    mailService.sendMail(to, subject, body);


    return "redirect:/boards/list";
  }
```

▼ C R U D(댓글 등록)

BoardController.java

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

BoardService.java

```
void replyNewBoard(Map map);
```

BoardServiceImpl.java

```
@Override
  public void replyNewBoard(Map map) {
    boardRepository.replyNewBoard(map);

  }
```

BoardRepository.java

```
void replyNewBoard(Map map);
```

BoardRepositoryImpl.java

```
@Override
  public void replyNewBoard(Map map) {
    this.sqlSessionTemplate.insert("board.insert_reply", map);
  }
```

board_SQL.xml

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

board.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib uri="http://www.springframework.org/security/tags" prefix="sec" %>
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+8O0xUf+/Im1MZjXxxgOcBQBXU=" crossorig

<head>
  <title>글 상세보기</title>
  <%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
      <meta charset="utf-8">

</head>
<body class="container">
<!-- 글 -->
<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.blike}]</p>
    <p class="btn btn-danger btn-sm">싫어요 [${board.bhate}]</p>
    </div>
    <div class="d-flex justify-content-end badge bg-light text-dark">${board.bdate}</div>
  </div>
</div>

<!-- 댓글 -->
```

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

  ${reply.bcontent}
  ${reply.bdate}
  ${reply.bwriter}
<br>


  </c:forEach>


  <!-- 댓글 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}" >
<textarea name="bcontent" id="bcontent" rows="5" class="form-control"></textarea>
<input type="button" class="btn btn-primary my-2" onclick="replyNewFunction()" value="댓글 등록">

  </div>
</div>





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

</body>

<iframe name="cart" style="display: none;"></iframe>

<script type="text/javascript">

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>




</html>
```

▼ C R U D(댓글 목록)

BoardController.java

```
@GetMapping("/board")
  public String requestCarById(@RequestParam("id") String bid, Model model) {
    //주 게시물
    Board boardById = boardService.getBoardById(bid);
    model.addAttribute("board", boardById);


    //답변 게시물
    List<Board> list = boardService.getReplyById(bid);
    int cnt = list.size();
    model.addAttribute("replyList", list);
    model.addAttribute("cnt", cnt);


    return "boards/board";
  }
```

BoardService.java

```
List<Board> getReplyById(String bid);
```

BoardServiceImpl.java

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

BoardRepository.java

```
List<Board> getReplyById(String bid);
```

BoardRepositoryImpl.java

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

board_SQL.xml

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

board.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib uri="http://www.springframework.org/security/tags" prefix="sec" %>
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+8O0xUf+/Im1MZjXxxgOcBQBXU=" crossorig

<head>
  <title>글 상세보기</title>
  <%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
      <meta charset="utf-8">
```

```
</head>
<body class="container">
<!-- 글 -->
<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.blike}]</p>
    <p class="btn btn-danger btn-sm">싫어요 [${board.bhate}]</p>
    </div>
    <div class="d-flex justify-content-end badge bg-light text-dark">${board.bdate}</div>
  </div>
</div>

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

  <div class="card">
  <div class="card-header">
    ${reply.bwriter}
  </div>
  <div class="card-body">
    <blockquote class="blockquote mb-0">
      <p>${reply.bcontent}</p>
      <footer class="blockquote-footer">${reply.bdate}</footer>
    </blockquote>
  </div>
</div>



<br>


  </c:forEach>


  <!-- 댓글 등록 -->
  <sec:authentication property="principal" var="user"/>
<input id="boardWriter" type="hidden" value="${board.bwriter}" />
<input id="bwriter" type="hidden" value="${user.username}" />
<input id="bid" type="hidden" name="bid" value="${board.bid}" >
<textarea name="bcontent" id="bcontent" rows="5" class="form-control"></textarea>
<input type="button" class="btn btn-primary my-2" onclick="replyNewFunction()" value="댓글 등록">

  </div>
</div>





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

</body>

<iframe name="cart" style="display: none;"></iframe>

<script type="text/javascript">

function replyNewFunction() {
  $.ajax({
    type:"POST",
    url:"/boards/replyNew",
    data:{
      boardWriter : document.getElementById('boardWriter').value,
      bid : document.getElementById('bid').value,
      bwriter : document.getElementById('bwriter').value,
      bcontent : document.getElementById('bcontent').value,
    },
    beforeSend : function(xhr)
        {   /*데이터를 전송하기 전에 헤더에 csrf값을 설정한다*/
            xhr.setRequestHeader("${_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>




</html>
```

▼ 댓글 등록시 이메일 전송처리

BoardController.java

```
@ResponseBody
  @RequestMapping("/replyNew")
  public void replyNew(@RequestParam Map<String, Object> map) {

    boardService.replyNewBoard(map);

    String boardWriter = (String) map.get("boardWriter");

    User user = userService.existUsername(boardWriter);

    String to = user.getUemail();
    String subject = (String) map.get("bwriter") + "님이 댓글을 등록하였습니다.";
    String body = (String) map.get("bcontent");

    mailService.sendMail(to, subject, body);

  }
```

board.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib uri="http://www.springframework.org/security/tags" prefix="sec" %>
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+8O0xUf+/Im1MZjXxxgOcBQBXU=" crossorig

<head>
  <title>글 상세보기</title>
  <%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
      <meta charset="utf-8">

</head>
<body class="container">
<!-- 글 -->
<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.blike}]</p>
    <p class="btn btn-danger btn-sm">싫어요 [${board.bhate}]</p>
    </div>
    <div class="d-flex justify-content-end badge bg-light text-dark">${board.bdate}</div>
  </div>
</div>

<!-- 댓글 -->
<div class="card">
  <div class="card-body">
  댓글 리스트
  <br><br>
```

```
<b>${cnt}개의 답변이 있습니다.</b><br>
  <c:forEach items="${replyList}" var="reply">

  <div class="card">
  <div class="card-header">
    ${reply.bwriter}
  </div>
  <div class="card-body">
    <blockquote class="blockquote mb-0">
      <p>${reply.bcontent}</p>
      <footer class="blockquote-footer">${reply.bdate}</footer>
    </blockquote>
  </div>
</div>



<br>


  </c:forEach>


  <!-- 댓글 등록 -->
  <sec:authentication property="principal" var="user"/>
<input id="boardWriter" type="hidden" value="${board.bwriter}" />
<input id="bwriter" type="hidden" value="${user.username}" />
<input id="bid" type="hidden" name="bid" value="${board.bid}" >
<textarea name="bcontent" id="bcontent" rows="5" class="form-control"></textarea>
<input type="button" class="btn btn-primary my-2" onclick="replyNewFunction()" value="댓글 등록">

  </div>
</div>





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

</body>

<iframe name="cart" style="display: none;"></iframe>

<script type="text/javascript">

function replyNewFunction() {
  $.ajax({
    type:"POST",
    url:"/boards/replyNew",
    data:{
      boardWriter : document.getElementById('boardWriter').value,
      bid : document.getElementById('bid').value,
      bwriter : document.getElementById('bwriter').value,
      bcontent : document.getElementById('bcontent').value,
    },
    beforeSend : function(xhr)
        {   /*데이터를 전송하기 전에 헤더에 csrf값을 설정한다*/
            xhr.setRequestHeader("${_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>





</html>
```

▼ 게시판 좋아요, 싫어요 기능추가

BoardController.java

```
@ResponseBody
  @RequestMapping("/check")
  public void check(@RequestParam Map<String, Object> map) {

    boardService.checkBoard(map);

  }
```

BoardService.java

```
void checkBoard(Map<String, Object> map);
```

BoardServiceImpl.java

```
@Override
  public void checkBoard(Map<String, Object> map) {
    boardRepository.checkBoard(map);
  }
```

BoardRepository.java

```
void checkBoard(Map<String, Object> map);
```

BoardRepositoryImpl.java

```
@Override
  public void checkBoard(Map<String, Object> map) {
    String check = (String) map.get("check");

    if(check.equals("blike")) {
      this.sqlSessionTemplate.update("board.update_like", map);
    }else {
      this.sqlSessionTemplate.update("board.update_hate", map);
    }



  }
```

board_SQL.xml

```
<update id="update_like" parameterType="hashMap">
<![CDATA[
UPDATE board SET blike = blike + 1 WHERE bid = #{bid}
  ]]>
</update>

<update id="update_hate" parameterType="hashMap">
<![CDATA[
UPDATE board SET bhate = bhate + 1 WHERE bid = #{bid}
  ]]>
</update>
```

board.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@ taglib uri="http://www.springframework.org/security/tags" prefix="sec" %>
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+8O0xUf+/Im1MZjXxxgOcBQBXU=" crossori

<head>
  <title>글 상세보기</title>
  <%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
      <meta charset="utf-8">

</head>
<body class="container">
<!-- 글 -->
<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 [${board.bview}]</p>
    <a href="javascript:checkFunction('${board.bid}', 'blike')" style="height: 50%;" class="btn btn-primary btn-sm">좋아요 [${board.
    <a href="javascript:checkFunction('${board.bid}', 'bhate')" style="height: 50%;" class="btn btn-danger btn-sm">싫어요 [${board.b
    </div>
    <div class="d-flex justify-content-end badge bg-light text-dark">${board.bdate}</div>
  </div>
</div>

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

  <div class="card">
  <div class="card-header">
    ${reply.bwriter}
  </div>
  <div class="card-body">
    <blockquote class="blockquote mb-0">
      <p>${reply.bcontent}</p>
      <footer class="blockquote-footer">${reply.bdate}</footer>
    </blockquote>
  </div>
</div>



<br>


  </c:forEach>


  <!-- 댓글 등록 -->
  <sec:authentication property="principal" var="user"/>
<input id="boardWriter" type="hidden" value="${board.bwriter}" />
<input id="bwriter" type="hidden" value="${user.username}" />
<input id="bid" type="hidden" name="bid" value="${board.bid}" >
<textarea name="bcontent" id="bcontent" rows="5" class="form-control"></textarea>
<input type="button" class="btn btn-primary my-2" onclick="replyNewFunction()" value="댓글 등록">

  </div>
</div>





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

</body>

<iframe name="cart" style="display: none;"></iframe>

<script type="text/javascript">

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

  })

  window.location.reload();
}

function checkFunction(bid, check) {
  $.ajax({
    type:"POST",
    url:"/boards/check",
    data:{
      bid : bid,
      check : check
    },
    beforeSend : function(xhr)
        {   /*데이터를 전송하기 전에 헤더에 csrf값을 설정한다*/
            xhr.setRequestHeader("${_csrf.headerName}", "${_csrf.token}");
        },
    success:function(result) {
      window.location.reload();
    },
    error:function(request, status, error) {
      alert(request.status + " " + request.responseText);
    }

  })


}

</script>




</html>
```

▼ 게시판에 댓글목록 뜨는 오류 해결하기

board_SQL.xml

```
<select id="select_list" resultType="com.carshop.board.Board">
<![CDATA[
  SELECT * FROM board WHERE bpid IS NULL ORDER BY bid DESC
  ]]>
</select>
```

▼ 게시판 조회수 기능추가

BoardController.java

```
@GetMapping("/board")
  public String requestCarById(@RequestParam("id") String bid, Model model) {

    //view 데이터를 불러오기 전에 조회수 증가
    Map<String, Object> map = new HashMap<String, Object>();
    map.put("bid", bid);
    map.put("check", "bview");
    boardService.checkBoard(map);

    //주 게시물
    Board boardById = boardService.getBoardById(bid);
    model.addAttribute("board", boardById);


    //답변 게시물
    List<Board> list = boardService.getReplyById(bid);
    int cnt = list.size();
    model.addAttribute("replyList", list);
    model.addAttribute("cnt", cnt);



    return "boards/board";
  }
```

BoardRepositoryImpl.java

```
@Override
  public void checkBoard(Map<String, Object> map) {
    String check = (String) map.get("check");

    if(check.equals("blike")) {
      this.sqlSessionTemplate.update("board.update_like", map);
    }else if(check.equals("bview")) {
      this.sqlSessionTemplate.update("board.update_view", map);
    }else {
      this.sqlSessionTemplate.update("board.update_hate", map);
    }



  }
```

board_SQL.xml

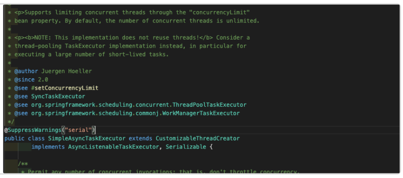
```
<update id="update_view" parameterType="hashMap">
<![CDATA[
UPDATE board SET bview = bview + 1 WHERE bid = #{bid}
  ]]>
</update>
```

@Async[비동기 메소드 작성 어노테이션]

[Spring] @Async Annotation(비동기 메소드 사용하기)

java.util.concurrent.ExecutorService을 활용해서 비동기 방식의 method를 정의 할 때마다, 위와 같이 Runnable의 run()을 재구현해야 하는 등 동일한 작업들의 반복이 잦았다. With @Async @Async Annotati

https://velog.io/@gillog/Spring-Async-Annotation비동기-메소드-사용하기



```
 * <p>Supports limiting concurrent threads through the "concurrencyLimit"
 * bean property. By default, the number of concurrent threads is unlimited.
 *
 * <p><b>NOTE: This implementation does not reuse threads!</b> Consider a
 * thread-pooling TaskExecutor implementation instead, in particular for
 * executing a large number of short-lived tasks.
 *
 * @author Juergen Hoeller
 * @since 2.0
 * @see #setConcurrencyLimit
 * @see SyncTaskExecutor
 * @see org.springframework.scheduling.concurrent.ThreadPoolTaskExecutor
 * @see org.springframework.scheduling.commonj.WorkManagerTaskExecutor
 */
@SuppressWarnings("serial")
public class SimpleAsyncTaskExecutor extends CustomizableThreadCreator
		implements AsyncListenableTaskExecutor, Serializable {

	/**
	 * Permit any number of concurrent invocations: that is, don't throttle concurrency.
```

@RequestBody & @ResponseBody

@ReqeustBody와 @ResponseBody 언제 사용할까?

비동기 통신을 위한 어노테이션

https://medium.com/webeveloper/reqeustbody와-responsebody-언제-사용할까-2efcab364edb