

**BLOB**

## 공통 적용 처리

---

### ❖ 브라우저 측 컨텍스트 경로 초기 설정

- `<c:url value="/" var="root"/>`
  - 브라우저 최초 접속시 JsessionID 쿠키 값이 설정되는 문제 있음
- EL 내장 객체를 이용하여 컨텍스트 경로 추출하여 root 속성 설정(request 스코프)

```
<c:set value="${pageContext.request.contextPath}"/"  
      var="root" scope="request"/>
```

## 공통 적용 처리

### ❖ resource/js/common.js

- o Date 객체에 날짜 출력 문자열 포매팅 prototype 메서드 정의

```
Date.prototype.toString = function() {  
    var year = this.getFullYear();  
    var month = this.getMonth() + 1;  
    month = month < 10 ? '0' + month : month;  
    var date = this.getDate();  
    date = date < 10 ? '0' + date : date;  
    return year + '-' + month + '-' + date;  
}
```

```
Date.prototype.toDatetime = function() {  
    var hour = this.getHours();  
    hour = hour < 10 ? '0' + hour : hour;  
    var minute = this.getMinutes();  
    minute = minute < 10 ? '0' + minute : minute;  
    var second = this.getSeconds();  
    second = second < 10 ? '0' + second : second;  
  
    return this.toString() + ' ' +  
        hour + ':' + minute + ':' + second;  
}
```

## 공통 적용 처리

---

### ❖ default\_template.jsp

```
        :  
<c:set value="${pageContext.request.contextPath}/"  
        var="root" scope="request"/>  
<link rel="stylesheet"  
      href="<c:url value="/resources/css/main.css"/>" />  
<script src="<c:url value="/resources/js/common.js"/>"></script>  
<script>  
$(function(){  
  
  });  
</script>  
</head>  
  
<body>  
  :
```

## ❖ 아바타 크기 조정용 ImageUtil

○ edu.iot.butter.util 패키지

```
public class ImageUtil {  
    public static byte[] makeThumb(byte [] data) throws Exception {  
        return makeThumb(data, 100, 100);  
    }  
  
    public static byte[] makeThumb(byte [] data, int width, int height)  
        throws Exception {  
        try(  
            ByteArrayInputStream in = new ByteArrayInputStream(data);  
            ByteArrayOutputStream out = new ByteArrayOutputStream();  
        ){  
            Thumbnails.of(in)  
                .size(width,height)  
                .crop(Positions.CENTER)  
                .toOutputStream(out);  
            return out.toByteArray();  
        }  
    }  
}
```

# BLOB - Avata

---

## ❖ AVATAS 테이블 정의

```
CREATE TABLE AVATAS(  
    user_id VARCHAR2(20) PRIMARY KEY,  
    image BLOB  
);
```

# BLOB - Avata

---

## ❖ 모델

```
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Avata {
    private String userId;
    private byte[] image;
}
```

# BLOB – Avata

---

## ❖ 매퍼 : resources/mapper/avata-mapper.xml

```
<?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="edu.iot.butter.dao.AvataDao">
    <select id="getCount" resultType="int">
        SELECT count(*)
        FROM avatas
    </select>

    <select id="selectList" parameterType="Pagination"
        resultType="Avata">
        SELECT user_id, image
        FROM (
            SELECT ROW_NUMBER() OVER (ORDER BY user_id) AS SEQ,
                user_id, image
            FROM avatas)
        WHERE SEQ BETWEEN #{start} AND #{end}
    </select>
```



# BLOB – Avata

---

## ❖ 매퍼 : resources/mapper/avata-mapper.xml

```
<select id="selectOne" parameterType="int" resultType="Avata">
    SELECT * FROM avatas
    WHERE user_id=#{user_id}
</select>
```

```
<insert id="insert" parameterType="Avata">
    INSERT INTO avatas (user_id, image)
    values("#{userId}, #{image})
</insert>
```

```
<update id="update" parameterType="Avata">
    UPDATE avatas SET
        image = #{image}
    WHERE user_id=#{userId}
</update>
```

```
<delete id="delete" parameterType="Avata">
    DELETE avatas WHERE user_id=#{userId}
</delete>
```

```
</mapper>
```

## BLOB – Avata

---

### ❖ Dao : AvataDao

```
public interface AvataDao extends BaseDao<Avata, String>{  
  
}
```

# BLOB - Avata

---

## ❖ 서비스 수정: MemberService

```
public interface MemberService {  
    :  
    // 아바타 처리  
    byte[] getAvata(String userId) throws Exception;  
  
    boolean insertAvata(Avata avata)throws Exception;  
  
    boolean updateAvata(Avata avata)throws Exception;  
  
    boolean deleteAvata(String userId) throws Exception;  
}
```

# BLOB - Avata

---

## ❖ 서비스 : MemberServiceImpl

```
@Autowired
AvataDao avataDao;
// 아바타 처리
@Override
public byte[] getAvata(String userId) throws Exception {
    Avata avata = avataDao.selectOne(userId);
    if(avata==null) {    // 아바타 이미지가 없는 경우
        // 익명 사용자 미리 하나 등록 해둠
        avata = avataDao.selectOne("anonymous");
    }

    return avata.getImage();
}

@Override
public boolean insertAvata(Avata avata) throws Exception {
    avata.setImage(ImageUtil.makeThumb(avata.getImage()));
    return avataDao.insert(avata) == 1;
}
```

## BLOB - Avata

---

### ❖ 서비스 : MemberServiceImpl

```
@Override
public boolean updateAvata(Avata avata) throws Exception {
    avata.setImage(ImageUtil.makeThumb(avata.getImage()));
    if(avataDao.update(avata) == 1) {
        return true;
    } else {
        // 존재하지 않는 경우
        return avataDao.insert(avata) == 1;
    }
}

@Override
public boolean deleteAvata(String userId) throws Exception {
    return avataDao.delete(userId) == 1;
}
```

# BLOB - Avata

---

## ❖ 컨트롤러 : ProfileController

- 아바타 이미지 보내기 : /butter/member/avata?userId=admin

```
@RequestMapping(value="/avata", method=RequestMethod.GET)
public ResponseEntity<byte[]> getAvata(
    @RequestParam("userId") String userId) throws Exception {

    final HttpHeaders headers = new HttpHeaders();
    headers.setContentType(MediaType.IMAGE_JPEG);
    return new ResponseEntity<byte[]>(service.getAvata(userId),
        headers, HttpStatus.OK);
}
```

## BLOB - Avata

❖ 회원 가입, 수정 뷰 : `account/join.jsp`, `member/profile.jsp`

```
<form:form commandName="member" enctype="multipart/form-data" >
    :

    <div class="md-form">
        <label for="name">이름</label>
        <form:input path="name" class="form-control" required/>
    </div>

    <div class="mb-3">
        
        <label for="avata">아바타 이미지</label>
        <input type="file" name="avata"/>
    </div>
    :
```

# BLOB - Avata

## ❖ 컨트롤러 : AccountController

- 회워 가입시 아바타 등록 처리

```
@RequestMapping(value="/join", method=RequestMethod.POST)
public String joinSubmit(@Valid Member member,
                        BindingResult result,
                        @RequestParam("avata") MultipartFile mFile,
                        RedirectAttributes ra) throws Exception {
    if(result.hasErrors()) return "account/join";
    service.create(member);

    // 아바타 저장
    if(mFile != null && !mFile.isEmpty()) {
        service.insertAvata(new Avata(member.getUserId(),
                                      mFile.getBytes()));
    }

    ra.addFlashAttribute("member", member);
    return "redirect:/join_success";
}
```



# BLOB - Avata

## ❖ 컨트롤러 : ProfileController

- 회원 정보 수정시 아바타 수정 처리

```
@RequestMapping(value="/edit", method=RequestMethod.POST)
public String editSubmit(@Valid Member member,
                        BindingResult result,
                        @RequestParam("avata") MultipartFile mFile,
                        HttpSession session) throws Exception {
    if(result.hasErrors()) return "member/edit";

    if(!service.update(member)) {// 실패한 경우
        result.reject("updateFail", "비밀번호가 일치하지 않습니다.");
        return "member/edit";
    }
    // 아바타 수정
    if(mFile != null && !mFile.isEmpty()) {
        service.updateAvata(new Avata(member.getUserId(),
                                       mFile.getBytes()));
    }
    session.setAttribute("USER", member);
    return "redirect:profile";
}
```

## BLOB - Avata

---

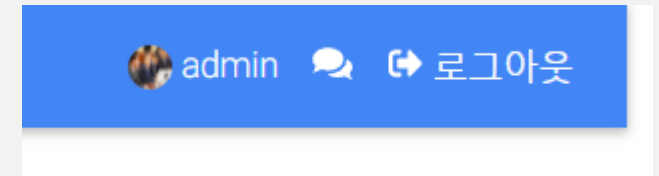
### ❖ 회원 정보 보기 : member/profile.jsp

```
    :  
<div class="row">  
    <div class="col-md-2">이름</div>  
    <div class="col-md-10">${member.name}</div>  
</div>  
  
<div class="row">  
    <div class="col-md-2">아바타</div>  
    <div class="col-md-10">  
          
    </div>  
</div>
```

# BLOB - Avata

## ❖ 메뉴 : menu.jsp

```
<li class="nav-item">
  <a class="nav-link" href="${root}member/profile">
    
    ${USER.userId}
  </a>
</li>
```



# BLOB - Avata

## ❖ 인터셉터 설정

```
<interceptors>
  <interceptor>
    <mapping path="/admin/**" />
    <mapping path="/talk/*" />
    <mapping path="/gallery/*" />
    <mapping path="/member/*" />
    <mapping path="/board/create" />
    <exclude-mapping path="/member/avata" />
    <beans:ref bean="loginInterceptor" />
  </interceptor>
  <interceptor>
    <mapping path="/admin/**" />
    <beans:ref bean="adminInterceptor" />
  </interceptor>
</interceptors>
```