BLOB

BLOB - Avata

❖ 아바타 크기 조정용 ImageUtil

```
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 AVATA(
   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

❖ mapper/avata-mapper.xml

```
<mapper namespace="edu.iot.common.dao.AvataDao">
   <select id="findById"  parameterType="String" resultType="Avata">
      SELECT * FROM avata
      WHERE user_id=#{userId}
   </select>
   <insert id="insert" parameterType="Avata">
      INSERT INTO avata (user_id, image)
      values(#{userId}, #{image})
   </insert>
   <update id="update" parameterType="Avata">
      UPDATE avata SET
         image = #{image}
      WHERE user_id=#{userId}
   </update>
   <delete id="delete" parameterType="String">
      DELETE FROM avata WHERE user_id=#{userId}
   </delete>
</mapper>
```

BLOB – Avata

❖ **Dao : AvataDao**

```
public interface AvataDao extends CrudDao<Avata, String>{

}
```

BLOB - Avata

❖ 서비스 수정: MemberService

```
public interface MemberService {
   :

   // 아바타 처리
   byte[] getAvata(String userId) throws Exception;

   void insertAvata(Avata avata)throws Exception;

   void updateAvata(Avata avata)throws Exception;

   void 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.findById(userId);
   if(avata==null) {    // 아바타 이미지가 없는 경우
      // 익명 사용자 미리 하나 등록 해둠
      avata = avataDao.findById("anonymous");
   }

   return avata.getImage();
}

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

BLOB - Avata

❖ 서비스 : MemberServiceImpl

```
@Override
public void updateAvata(Avata avata) throws Exception {
   avata.setImage(ImageUtil.makeThumb(avata.getImage()));

   if(avataDao.update(avata) == 0) {  // 존재하지 않는 경우
      avataDao.insert(avata);
   }
}

@Override
public void deleteAvata(String userId) throws Exception {
   avataDao.delete(userId);
}
```

BLOB - Avata

❖ 회원 가입 : account/join.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"/>
      </div>

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

BLOB - Avata

❖ 컨트롤러 : AccountController

- 회원 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.join(member);

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

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

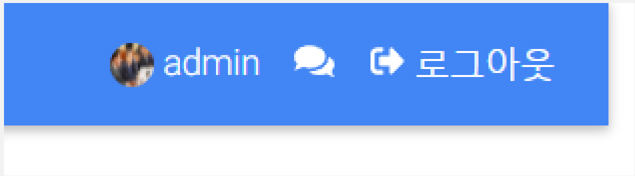
❖ account/join_success.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib tagdir="/WEB-INF/tags/util" prefix="iot"%>

<iot:page-header icon="user" title="회원 가입 완료"/>

${member.userId}님 회원 가입이 완료됐습니다.
```

BLOB - Avata

❖ 메뉴 : menu.jsp

```
<li class="nav-item">
   <a class="nav-link" href="${context}/member/view">
      <img src="${context}/member/avata/${USER.userId}"
         class="rounded-circle avata-sm" >
      ${USER.userId}
   </a>
</li>
```



BLOB – Avata

❖ **main.css**

```
.avata-sm {
   width:25px;
   height:25px;
}

.avata-md {
   width:50px;
   height:50px;
}

.avata-lg {
   width:75px;
   height:75px;
}
```

BLOB - Avata

❖ 컨트롤러 : MemberController(1)

- 아바타 이미지 내보내기 : /member/avata/{userId}

```
@ResponseBody
@RequestMapping(value="/avata/{userId}",
            method=RequestMethod.GET,
            produces = "image/jpeg")
public byte[] getAvata(
      @PathVariable("userId") String userId) throws Exception {

   return service.getAvata(userId);

}
```

BLOB – Avata

❖ 컨트롤러 : MemberController(2)

- 아바타 이미지 내보내기 : /member/avata/{userId}

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

   final HttpHeaders headers = new HttpHeaders();   // 응답헤더

   try {
      byte[] avata = service.getAvata(userId);
      headers.setContentType(MediaType.IMAGE_JPEG); // Mimetype
      return new ResponseEntity<byte[]>(avata, headers,
                                HttpStatus.OK);     // 200 응답코드
   } catch(Exception e) {
      return new ResponseEntity<byte[]>(null, headers,
                 HttpStatus.INTERNAL_SERVER_ERROR); // 500 응답코드
   }
}
```

BLOB - Avata

- **MemberController**

```
@RequestMapping(value="/view", method=RequestMethod.GET)
public void view() throws Exception {
}
```

BLOB - Avata

❖ 회원 정보 보기 : member/view.jsp

```
<h1 class="my-5">회원 정보 보기</h1>

<div class="row">
   <div class="col-md-2">사용자 ID</div>
   <div class="col-md-10">
      <img src="avata/${USER.userId}"
         class="rounded-circle avata-sm" ><br>
      ${USER.userId}
   </div>
</div>
   :

<div class="row mt-3">
   <div class="col-md-2"></div>
   <div class="col-md-10">
      <a href="edit" class="btn btn-primary btn-sm">수정</a>
   </div>
</div>
```

BLOB - Avata

❖ MemberController

```
@RequestMapping(value="/edit", method=RequestMethod.GET)
public void editForm(HttpSession session,
                  Model model) throws Exception {
   model.addAttribute("member", session.getAttribute("USER"));
}
```

BLOB – Avata

❖ 회원 정보 보기 : member/edit.jsp

```
<h1 class="my-5">회원 정보 수정</h1>

<form:form commandName="member"  enctype="multipart/form-data">
   <form:hidden path="userId" />
   <form:hidden path="name" />
   <div>
      <label>사용자 ID</label>
      <img src="avata/${USER.userId}"
         class="rounded-circle avata-sm" >
      <p>${member.userId}</p>
   </div>

   <div>
      <label for="name">이름</label>
      <p>${member.name}</p>
   </div>
   <div>
      <label>아바타</label>
      <p><input type="file" name="avata"/></p>
   </div>
```

BLOB - Avata

❖ 컨트롤러 : MemberController

- 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

❖ 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>
```