Gallery
- 파일 업로드 -

❖ 이미지 추가 폼(update.jsp)

```
<h4>이미지 파일 추가</h4>

<form action="add_image" method="post"
      enctype="multipart/form-data">
   <input type="hidden" name="galleryId"
          value="${gallery.galleryId}">

   <div class="md-form">
      <label>타이틀</label>
      <input type="text" name="title" class="form-control">
   </div>

   <div class="md-form">
      <label>출처</label>
      <input type="text" name="source" class="form-control">
   </div>

   <input type="file" name="file">
```

❖ 이미지 추가 폼(update.jsp)

```
   <p>
      <button type="submit" class="btn btn-primary">
         <i class="fas fa-check"></i> 확인
      </button>
      <button type="reset" class="btn btn-success">
         <i class="fas fa-undo"></i> 취소
      </button>
   </p>
</form>
```

파일 업로드

❖ 폼 인코딩

- enctype="application/x-www-form-urlencoded"시(디폴트) body 내용

```
title=배트맨&description=배트맨 오토바이&…
```

- binary 데이터 전송 불가

파일 업로드

❖ 폼 인코딩

- enctype="multipart/form-data"시 body 내용

```
------WebKitFormBoundaryJZ5taSn15dZXBCpl
Content-Disposition: form-data; name="title"

배트맨
------WebKitFormBoundaryJZ5taSn15dZXBCpl
Content-Disposition: form-data; name="description"

배트맨 오토바이트
------WebKitFormBoundaryJZ5taSn15dZXBCpl--

   :
```

Part

Part

파일 업로드

❖ 파일 업로드 의존성 설정

- Commons Fileupload 1.3.1
- Commons IO 2.4

```
<!-- Apache Commons file upload -->
<dependency>
   <groupId>commons-fileupload</groupId>
   <artifactId>commons-fileupload</artifactId>
   <version>1.3.1</version>
</dependency>
<!-- Apache Commons IO -->
<dependency>
   <groupId>commons-io</groupId>
   <artifactId>commons-io</artifactId>
   <version>2.4</version>
</dependency>
```

파일 업로드

❖ MultipartResolver 설정

- MultipartResover를 스프링 설정 파일에 등록
  - Multipart로 전달된 파일 처리 지원

- extension-context.xml

```
<bean id="multipartResolver"
   class="org.springframework.web.multipart.commons.CommonsMultipartResolver" />
```

파일 업로드

❖ MultipartResolver 설정

- MultipartResover를 스프링 설정 파일에 등록
  - CommonsMultipartResolver의 프로퍼티

프로퍼티	타입	설명
maxUploadSize	long	최대 업로드 가능한 바이트 크기. -1은 제한이 없음을 의미. 기본값은 -1
maxInMemorySize	int	디스크에 임시 파일을 생성하기 전에 메모리에 보관할 수 있는 최대 바이트 크기. 기본 값은 1024바이트
defaultEncoding	String	요청을 파싱할 때 사용할 캐릭터 인코딩. 지정하지 않을 경우, HttpServletRequest.setCharacterEncoding 메서드로 지정한 캐릭터 셋을 사용.

파일 업로드

❖ MultipartFile 인터페이스의 주요 메서드

- 단일 파일 업로드 처리시 사용

```
@RequestMapping(value="/update/add_image",
            method=RequestMethod.POST)
public String addImageSubmit(GalleryImage image,
      @RequestParam("file") MultipartFile imageFile)
      throws Exception {

   // 파일 업로드 및 db 저장

   return "redirect:/gallery/update/" + image.getGalleryId();
}
```

파일 업로드

❖ MultipartFile 인터페이스의 주요 메서드

메서드	설명
`String getName( )`	파라미터 이름 리턴
`String getOriginalFilename( )`	업로드 한 파일 이름 리턴
`boolean isEmpty( )`	업로드 한 파일이 존재하지 않는 경우 true 리턴
`long getSize( )`	업로드 한 파일의 크기 리턴
`byte[ ] getBytes( )` `throws IOException`	업로드한 파일 데이터의 바이트 배열리턴
`InputStream getInputStream( )` `throws IOException`	업로드 한 파일의 InputStream 리턴 InputStream의 사용 후 close( ) 호출 필요
`void transferTo(File dest)` `throws IOException`	업로드 한 파일 데이터를 지정한 파일로 저장

파일 업로드

❖ MultipartFile 인터페이스의 주요 메서드

- 단일 파일 업로드 처리시 사용

```
private void uploadImage(GalleryImage image,
                         MultipartFile imageFile) throws Exception {
   // 원본 이미지 파일명
   String fname = imageFile.getOriginalFilename();
   // 서버에서의 이미지 파일명
   long timestamp = System.currentTimeMillis();
   String serverFileName = timestamp + "_" + fname;

   // 업로드 파일 재배치
   File dest = new File("c:/temp/upload/" + serverFileName);
   imageFile.transferTo(dest);

   // GalleryImage 정보 설정
   image.setFileName(fname);
   image.setServerFileName(serverFileName);
}
```

파일 업로드

❖ MultipartFile 인터페이스의 주요 메서드

- 단일 파일 업로드 처리시 사용

```
@RequestMapping(value="/update/add_image",
            method=RequestMethod.POST)
public String addImageSubmit(GalleryImage image,
      @RequestParam("file") MultipartFile imageFile)
      throws Exception {

   // 파일 업로드
   uploadImage(image, imageFile);
   // db 저장

   return "redirect:/gallery/update/" + image.getGalleryId();
}
```

❖ GalleryImage DB 저장

- gallery-mapper.xml

```
<insert id="addImage" parameterType="GalleryImage"><![CDATA[
   insert into gallery_image (
      image_id, title, file_name, server_file_name,
      source, gallery_id)
   values(
      gallery_image_seq.nextval,
      #{title}, #{fileName}, #{serverFileName},
      #{source}, #{galleryId})
]]></insert>
```

파일 업로드

❖ GalleryDao 인터페이스

```
public interface GalleryDao
      extends CrudDao<Gallery, Long> {

   void addImage(GalleryImage image) throws Exception;

}
```

파일 업로드

❖ GalleryService 인터페이스

```
public interface GalleryService {
    :

    void addImage(GalleryImage image) throws Exception;

}
```

파일 업로드

❖ GalleryServiceImpl

```
public interface GalleryServiceImpl {
   :

   @Override
   public void addImage(GalleryImage image) throws Exception {
      dao.addImage(image);

   }
}
```

파일 업로드

❖ GalleryController

```
@RequestMapping(value="/update/add_image",
                method=RequestMethod.POST)
public String addImageSubmit(GalleryImage image,
      @RequestParam("file") MultipartFile imageFile) throws Exception
{

   uploadImage(image, imageFile);
   service.addImage(image);

   return "redirect:/gallery/update/" + image.getGalleryId();
}
```

❖ Gallery 이미지 목록 얻기

- gallery-mapper.xml

```
<select id="getImages" parameterType="long"
                resultType="GalleryImage"><![CDATA[
   select * from gallery_image
   where gallery_id = #{galleryId}
]]></select>
```

파일 업로드

❖ GalleryDao 인터페이스

```
public interface GalleryDao
      extends CrudDao<Gallery, Long> {

   void addImage(GalleryImage image) throws Exception;

   List<GalleryImage> getImages(long galleryId) throws Exception;
}
```

파일 업로드

❖ GalleryServiceImpl

```
@Override
public Gallery findById(long galleryId) throws Exception {
   Gallery gallery = dao.findById(galleryId);
   List<GalleryImage> list = dao.getImages(galleryId);
   gallery.setList(list);
   return gallery;
}
```

❖ 갤러리 이미지 목록 출력

- update.jsp

```
<h4>이미지 파일 목록</h4>
<ul>
   <c:forEach var="image" items="${gallery.list}">
      <li>${image.fileName}</li>
   </c:forEach>
</ul>
```