

Google App Engine의 BlobstoreService를 활용해 파일 업로드하기

GAE에서 1MB 이상의 파일을 업로드하려면 BlobstoreService API를 활용해 가능하다. BlobstoreService API에 대한 기본적인 사용 방법은 [Blobstore Java API Overview](#)에서 확인할 수 있다. BlobstoreService API는 최대 2GB 크기의 파일을 업로드하는 것까지 지원하고 있다. [Blobstore Java API Overview](#)에서 제공하는 샘플을 활용해 기본적인 테스트를 완료하고 SLiPP에 적용하는 단계를 거쳐 개발했다. 먼저 BlobstoreService에 데이터를 저장하려면 BlobstoreService API를 활용해 URL을 생성해야 한다.

```
@Controller
@RequestMapping("/upload/file")
public class UploadFileController {
    private BlobstoreService blobstoreService =
        BlobstoreServiceFactory.getBlobstoreService();

    @RequestMapping("/form")
    public String uploadForm(ModelMap model) throws Exception {
        model.addAttribute("uploadUrl",
            blobstoreService.createUploadUrl("/upload/file"));
        return "support/upload";
    }
}
```

UploadUrl을 JSP(또는 Freemarker)에서 직접 생성하지 않고 Spring MVC의 Controller에서 생성하도록 변경했다.

```
<html>
<head>
<title>Sustainable People, Programming, Programmer</title>
</head>
<body>
    <form action="${uploadUrl}" method="post"
        enctype="multipart/form-data">
        <input type="file" name="uploadFile">
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

upload.ftl에서는 Spring MVC Controller에서 생성된 uploadUrl을 사용하도록 했으며, 하나의 파일이 업로드 가능하도록 구현했다.

```

@Controller
@RequestMapping("/upload/file")
public class UploadFileController {
    private BlobstoreService blobstoreService =
BlobstoreServiceFactory.getBlobstoreService();

    @RequestMapping(value="", method=RequestMethod.POST)
    public String upload(HttpServletRequest request) throws Exception {
        Map<String, BlobKey> blobs =
blobstoreService.getUploadedBlobs(request);
        BlobKey blobKey = blobs.get("uploadFile");

        return "redirect:/serve/file?key=" + blobKey.getKeyString();
    }
}

```

마지막으로 upload한 파일은 BlobstoreService를 활용해 위 예제 소스와 같이 업로드하는 것이 가능하다. GAE의 Blobstore는 개발자가 직접 파일을 업로드하는 것이 아니라 GAE의 BlobstoreService의 getUploadedBlobs()에 HttpServletRequest를 전달하면 API에서 파일을 업로드하고 파일에 대한 메타 정보를 데이터베이스에 저장한다.

BlobstoreService를 통해 업로드한 파일의 메타 정보는 BlobInfo를 통해 조회하는 것이 가능하다. 업로드한 파일에 대한 메타데이터는 __BlobInfo__ 테이블에서 관리되고 있다.

BlobstoreService API를 활용할 경우 여러 개의 파일 입력품을 가지는 경우 치명적인 버그가 있다. 여러 개의 파일 입력품을 활용해 파일을 업로드할 때 업로드할 파일을 선택하지 않았음에도 불구하고 메타데이터가 생성된다. 불필요한 쓰레기 데이터가 생성되는 것이다. 로컬에서 테스트하는 경우에는 업로드할 파일이 있는 경우에만 정상적으로 메타데이터가 생성되는데 반해 실 서비스를 할 때는 메타데이터가 생성되는 버그가 존재하는 것을 확인했다.