

# INNORIX WP

## Premature Ending 조치 가이드



# Contents

## I 조치 가이드

- |                           |       |      |
|---------------------------|-------|------|
| 1. Premature Ending 발생 원인 | ..... | I -5 |
| 2. Premature Ending 해결 방안 | ..... | I -6 |

## Premature Ending 에러 조치 가이드

I

# 조치 가이드

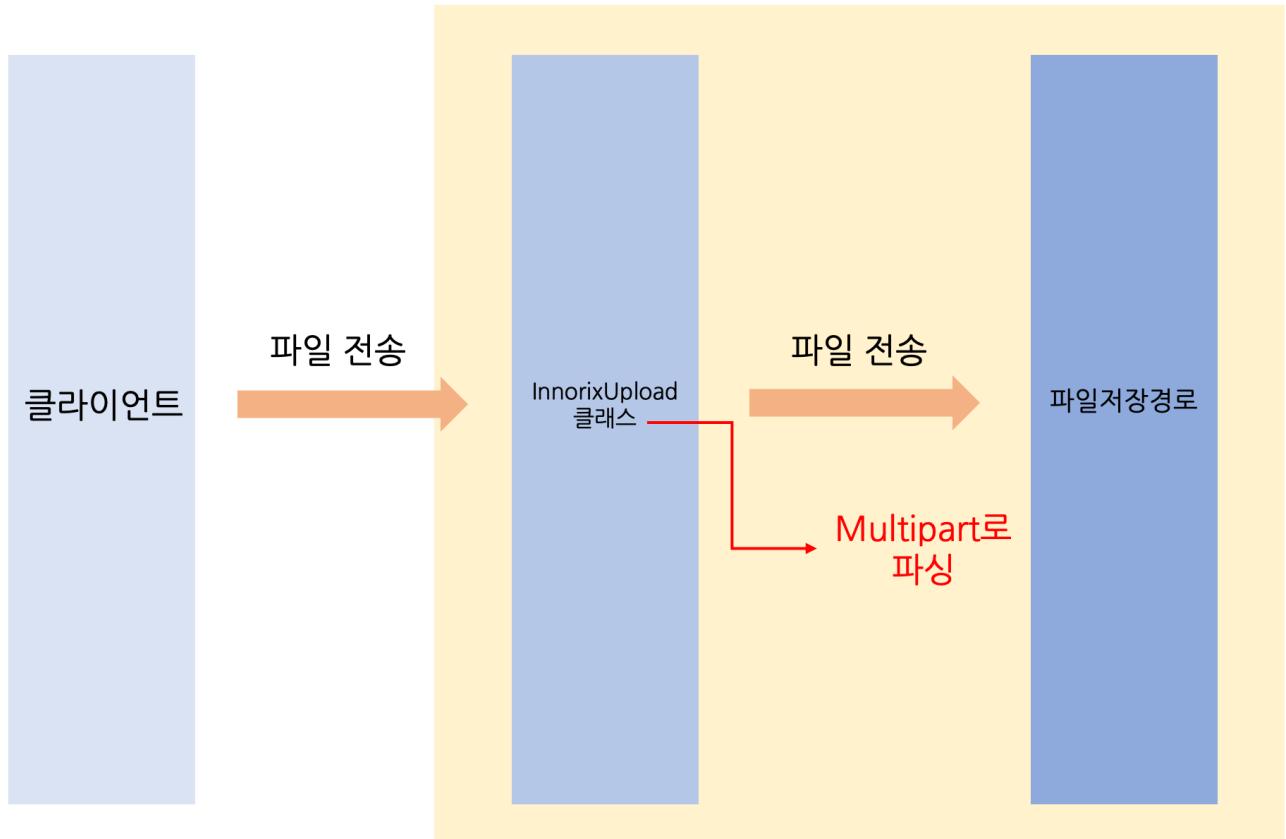
1. Premature Ending 발생 원인
2. Premature Ending 해결 방안





## Premature Ending 발생 이유

- INNORIX WP 업로드의 순차적인 흐름은 아래와 같습니다.



- 1) 클라이언트에서 파일 전송
- 2) InnorixUpload 클래스에서 Multipart로 파싱
- 3) 파일 저장

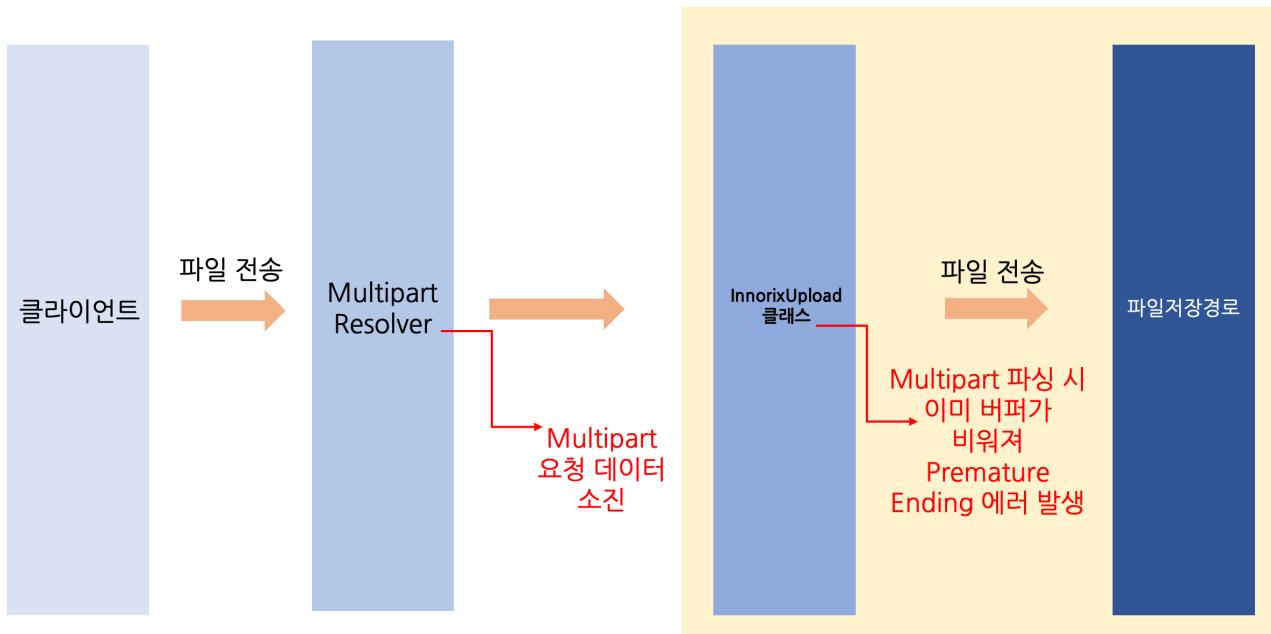


Premature Ending 에러의 경우,

클라이언트에서 파일 전송 후 InnorixUpload 클래스에서 Multipart로 파싱 하는 과정 사이에서

**특정 필터 또는 기능으로 인해 Multipart 요청 데이터가 소진되어 InnorixUpload 클래스에서**

**Multipart로 파싱하려는 시점에 이미 버퍼가 비워지는 이유로 Premature Ending 에러가 발생하게 됩니다.**



일반적으로 Spring 환경에서 파일 업로드를 구현하기 위해서는

Context 설정 파일 내에 CommonsMultipartResolver Bean을 등록하거나 Web.xml에 MultipartFilter 설정을 하게 됩니다.

이러한 설정으로 인하여, 모든 요청데이터는 InnorixUpload 클래스에서 Multipart로 파싱되는 과정이 진행되기 전에 소진됩니다.

- CommonsMultipartResolver 설정 예시 -> ex) servlet-context.xml(파일명은 다를 수 있습니다.)

```
<beans:bean id="filterMultipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <beans:property name="maxUploadSize" value="104857600"/>
</beans:bean>
```

- MultipartFilter 설정 예시 -> web.xml

```
<filter>
    <filter-name>multipartFilter</filter-name>
    <filter-class>org.springframework.web.multipart.support.MultipartFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>multipartFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```



## Premature Ending 에러 해결 방안

### Premature ending 해결 방안

근본적으로 문제가 되는 CommonsMultipartResolver의 Bean과 이노릭스는 아래와 같이 충돌이 발생하여 Premature Ending 에러가 발생하게 됩니다.

```
java.io.IOException: Corrupt form data: premature ending
    at com.innorix.multipartrequest.MultipartParser.<init>(MultipartParser.java:207)
    at com.innorix.multipartrequest.MultipartRequest.<init>(MultipartRequest.java:311)
    at com.innorix.multipartrequest.MultipartRequest.<init>(MultipartRequest.java:257)
    at com.innorix.transfer.InnorixUpload.init(InnorixUpload.java:865)
    at com.innorix.transfer.InnorixUpload.<init>(InnorixUpload.java:767)
```

이러한 문제를 해결하기 위해서는 Bean 설정을 제거하면 간단하나  
시스템의 구조상 병행하여 사용하는 경우가 생기게 되어  
INNORIX Upload Url로 전달되는 Servlet에 대해서는 해당 설정이 적용되지 않게  
Servlet 분기처리를 진행 해줘야 합니다.

※ 이후에 나오는 파일명은 예시로 지정된 이름으로 현재 사용하고 계시는 파일명과  
다를 수 있는 점 참고 부탁드립니다.

※ 파일명은 현재 사용하시고 계시는 환경에 맞게 설정해주시면 됩니다.



## Premature Ending 에러 해결 방안

### ◆ 분기처리 방법

- INNORIX Upload URL로 전달되는 Servlet 분기처리를 위해 innorix-context.xml 파일을 생성해줍니다.
- 생성된 innorix-context.xml 파일 내에서 InnorixController(업로드 시 사용 되는 소스가 들어있는 컨트롤러)가 사용될 수 있도록 해당 패키지를 지정해줍니다.  
(ex. com.innorix.gnit)

Screenshot of an IDE showing multiple files: web.xml, InnorixController.java, servlet-context.xml, innorix-context.xml, ViewController.java, combineSessionView.jsp, downloadView.jsp, and WebSessionListener.java. The innorix-context.xml file is open, showing configuration for static resources and view resolution. A red box highlights the <context:component-scan base-package="com.innorix.gnit" /> line, which is the part being discussed.

```

14
15    <!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resources in the ${webappRoot}/resources directory -->
16    <resources mapping="/**" location="/" />
17
18    <!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF/views directory -->
19    <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
20        <beans:property name="prefix" value="/WEB-INF/views/" />
21        <beans:property name="suffix" value=".jsp" />
22    </beans:bean>
23
24    <!-- -->
25    <context:component-scan base-package="com.innorix.gnit" />
26
27 </beans:beans>
28

```

- web.xml에서는 INNORIX Upload URL로 전달되는 Servlet에 대해서는 Multipart 설정이 적용되지 않도록 Servlet을 분기처리 해줍니다.
    - 호출되는 URL의 확장자가 \*.inno거나 URL 경로 앞부분에 /innorix/가 존재할 경우 Multipart가 동작하지 않도록 설정해줍니다.
- (확장자 및 URL 앞부분의 경로는 예시로 설정된 부분이며 환경에 따라 다르게 설정해주시면 됩니다.)

Screenshot of an IDE showing the web.xml file. A red box highlights the <url-pattern>/innorix/\*</url-pattern> line, indicating where the multipart processing is disabled for URLs starting with /innorix/.

```

15<!--
16    <servlet-mapping>
17        <servlet-name>appServlet</servlet-name>
18        <url-pattern>/</url-pattern>
19    </servlet-mapping>
20
21<!-- innorix servlet를 따로 분리하고 multipart 동작을 하지 않도록 함 -->
22<servlet>
23    <servlet-name>innorix</servlet-name>
24        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
25        <init-param>
26            <param-name>contextConfigLocation</param-name>
27            <param-value>/WEB-INF/spring/appServlet/innorix-context.xml</param-value>
28        </init-param>
29        <load-on-startup>1</load-on-startup>
30    </servlet>
31
32<servlet-mapping>
33    <servlet-name>innorix</servlet-name>
34    <url-pattern>/innorix/*</url-pattern>
35    <!-- <url-pattern>*.inno</url-pattern> // 모두 동일한 URL 사용시 분기처리에 어려움이 있을 수 있으므로 확장자로도 분기처리 가능 -->
36</servlet-mapping>

```



## Premature Ending 에러 해결 방안

### ◆ 분기처리 방법

- 페이지 화면이 출력 되는 경로( 및 InnorixController 외 경로)는 innorix-context.xml이 적용되지 않도록 해당 패키지들은 servlet-context.xml에 지정해줍니다.

```
web.xml      servlet-context.xml
22           <!-- multipart resolver -->
23           <beans:bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
24             <beans:property name="defaultEncoding" value="utf-8"/>
25             <beans:property name="maxUploadSize" value="9999999999"/>
26           </beans:bean>
27
28           <context:component-scan base-package="com.view.gnit" />
29
30
31           </beans:beans>
```

- 이미 별도의 MultipartResolver가 지정된 context파일이 존재할 경우, 해당 파일에서 InnorixController가 지정되지 않도록 이외의 패키지를 지정해줍니다.



## FAQ

### ◆ 자주 묻는 질문

#### 1. 분기처리를 진행하였음에도 불구하고 Premature Ending 에러가 계속 발생합니다.

1-1. servlet 분기처리 시 사용한 xml 파일 내 패키지 참조가 잘 되어 있는지 확인 부탁드립니다.

<url-pattern>을 Url Path로 설정하기 위해서는 동일한 패키지에 업로드를 구현하는 컨트롤러와 MultiPart를 사용하는 컨트롤러가 존재하여서는 안 되며 innorix-context.xml에서 업로드를 구현하는 컨트롤러의 패키지를 반드시 지정해 주셔야 합니다.

만약, Upload url의 경로가 www.abc.com/test/innorix/upload.do이고 www.abc.com/test 까지 공통으로 설정된 경로일 경우, URL path로의 분기처리가 어려울 수 있습니다.

위와 같이 url path처리가 어려울 경우, 'web.xml' 파일 내 분기처리 시 설정 된 <url-pattern>을 URL path가 아닌 별도의 확장자로 설정 후 확인 부탁드립니다.

추가로 해당 <url-pattern>을 확장자로 변경 시 매핑되는 upload Url도 변경해 주셔야 합니다.

1-2. 공통 filter로 설정된 multipartFilter가 있는지 확인 부탁드립니다.

filter 순서에서 전체 경로에 multipartFilter가 적용되어 있을 경우에는 innorix를 위한 servlet 분기처리가 진행되지 않습니다.

해당 상황의 경우에는 customFilter를 이용하여 해당 필터를 예외 처리 하여 이노릭스 Upload Url은 해당 Filter를 타지 않게 설정해 주셔야 합니다.

1-3. contextConfigLocation을 참조하는 XML 파일들 중 multipartResolver가 있는지 확인 부탁드립니다.

web.xml 파일 내 설정된

<param-value>classpath\*:egovframework/spring/com/context-\*.xml</param-value>에서 불러오는 xml 파일 중 multipartResolver가 설정되어 있는지 확인 부탁드립니다.

web.xml에서는 spring 설정 파일을 읽어오게 되어 있습니다.

스프링 설정파일을 읽어 들이기 위해서는 org.springframework.web.context.ContextLoaderListener 가 필요합니다.

(리스너가 동작해야 설정 파일을 읽어들이고 서블릿 이전에 서블릿을 초기화하는 용도로 필요한 객체들을 등록하고 만드는 것)

contextConfigLocation에서 설정된 xml 파일 중 MultipartResolver를 등록한 bean이 있다면 이노릭스 servlet 분기처리가 진행되지 않습니다.

해당 상황일 경우에는 multipartResolver를 등록한 Bean을 별도의 xml 파일에서 설정되게 이동 시킨 후 web.xml에서는 해당 파일을 읽어올 수 있게 처리해 됩니다.



설정 환경과 관련하여 전달 드리는 예제 파일은 모두 참고용으로,  
현재 사용하시는 설정 환경과 다를 수 있으며  
해당 예제 파일을 참고하시어 개발 환경 및 개발 요구에 맞춰 직접 수정하여 사용해 주셔야 합니다.

스프링 프레임워크 환경설정 관련하여 사용하시고 계시는 개발 환경에 맞게 예제를 수정하여  
모든 환경 설정을 수정해 드릴 수 없는 점 양해 부탁드리며,  
해당 설정 진행 시 현재 사용하시고 계시는 서버 환경에 대한 전체적인 파악이 필요하여  
개발사에 맞는 환경 설정을 직접 해드릴 수 없는 점 참고 부탁드리겠습니다.

해당 내용 관련하여 추가적으로 문의가 있으실 경우에는  
지엔기술 이노릭스 기술지원 메일 <[innorix@gnit.co.kr](mailto:innorix@gnit.co.kr)>로 회신주시면 확인 후 답변 드리도록 하겠습니다.