

- 실행환경 Migration 가이드
  - 업그레이드 방법
  - MultipartParsingResult 문제 해결
  - 기타 알 수 없는 충돌이 발생하는 경우

전자정부 표준프레임워크 2.0은 기존 표준프레임워크 1.0과의 호환성을 염두해두고 실행환경의 버전 업그레이드를 진행하였다. 따라서 표준프레임워크 1.0이 적용된 시스템에서도 표준프레임워크 2.0으로의 업그레이드를 실시할 수 있다. 다만, 각 프로젝트의 상황 및 환경에 따라서 버전 업그레이드에 따른 충돌, 버그, 불안정성이 발생할 수도 있으니 구축 및 운영 중인 프로젝트의 상황에 맞추어 사용하도록 권장한다.

## 업그레이드 방법

### 1. Maven을 사용하는 경우

Maven을 사용하는 경우 프로젝트에서 사용하는 pom.xml 파일을 수정하여 업그레이드를 진행할 수 있다. 기본적으로 표준프레임워크 실행환경에서 사용되는 오픈소스 라이브러리들만을 업그레이드 하고자 하는 경우 pom.xml 파일에서 egovframework.rte 패키지 아래에 있는 라이브러리들의 버전만 2.0으로 업그레이드 해주면 된다.

변경 전 (예)

```
<dependency>
  <groupId>egovframework.rte</groupId>
  <artifactId>egovframework.rte.psl.dataaccess</artifactId>
  <version>1.0.0</version>
</dependency>
```

변경 후 (예)

```
<dependency>
  <groupId>egovframework.rte</groupId>
  <artifactId>egovframework.rte.psl.dataaccess</artifactId>
  <version>2.0.0</version>
</dependency>
```

만약 표준프레임워크의 버전을 property를 사용하여 일괄적으로 관리하는 경우 property를 2.0.0으로 변경합니다. property 이름은 프로젝트에 따라 다를 수 있습니다.

변경 전 (예)

```
<properties>
  <egovframework.rte.version>1.0.0</spring.maven.artifact.version>
</properties>
```

변경 후 (예)

```
<properties>
  <egovframework.rte.version>2.0.0</spring.maven.artifact.version>
</properties>
```

표준프레임워크 실행환경은 각각의 실행환경들이 별도의 오픈소스 라이브러리와 dependency가 걸려있으므로, 모든 실행환경의 버전을 일괄적으로 업그레이드하는 것을 권장한다. 특정 실행환경 서비스의 업그레이드만 진행하여 1.0 버전과 2.0버전을 혼재하여 사용할 경우의 충돌이나 버그에 대해서는 테스트가 이루어지지 않았다.

혹은 실행환경 배포도구 플러그인을 사용하여 실행환경의 업그레이드를 자동으로 수행할 수 있다. 관련된 가이드는 [실행환경 배포도구]를 참고하여 주시기 바랍니다.

### 2. Maven을 사용하지 않는 경우

다운로드 메뉴에서 실행환경 라이브러리를 직접 다운로드 하여 사용한다. 이 경우 각각의 실행환경 서비스가 의존성을 가지고 있는 오픈소스 라이브러리를 자동으로 다운로드 받지 못하므로 수작업으로 필요한 라이브러리를 다운로드 받아야 한다. 위에 설명된 pom.xml 파일을 참고하여 의존성을 가지는 라이브러리들을 버전을 맞추어 다운로드 한다.

## MultipartParsingResult 문제 해결

만약 프로젝트내에서 spring-web의 MultipartParsingResult 클래스를 생성하는 경우, MultipartParsingResult의 생성자가 변경되어 컴파일 오류가 발생할 수 있습니다.

### Spring 2.x의 생성자

CommonsFileUploadSupport.MultipartParsingResult

```
public CommonsFileUploadSupport.MultipartParsingResult (Map multipartFiles, Map multipartParameters)
```

Create a new MultipartParsingResult.

- Parameters:multipartFiles - Map of field name to MultipartFile instance
- multipartParameters - Map of field name to form field String value

## Spring 3.x의 생성자

CommonsFileUploadSupport.MultipartParsingResult

```
public CommonsFileUploadSupport.MultipartParsingResult(MultiValueMap<String,MultipartFile> mpFiles,  
Map<String,String[]> mpParams)
```

Create a new MultipartParsingResult.

- Parameters:mpFiles - Map of field name to MultipartFile instance
- mpParams - Map of field name to form field String value

MultipartParsingResult를 생성하기 위하여 기존에 HashMap등의 형태로 인자를 전달했다면, MultiValueMap의 형태로 인자를 전달하여 인스턴스를 생성하도록 한다. MultiValueMap은 Map 인터페이스의 구현클래스다. 따라서 HashMap등의 인스턴스를 Map 형태로 선언하여 전달했을 경우, MultiValueMap으로 캐스팅하여 전달하는 형태로 해결 가능하다.

## 기타 알 수 없는 충돌이 발생하는 경우

프로젝트에 따라서 기존에 사용하던 라이브러리, 코드, WAS, DB 등과의 충돌이 발생할 수 있다. 알 수 없는 충돌이 발생하고 해결이 어려운 경우 표준프레임워크센터에 프로젝트의 환경 및 에러메시지를 포함하여 문의한다. WAS, DB 등에서 Spring 3.0을 지원하지 않는 경우 해결하기 어려울 수 있으니 Spring 2.5를 사용하는 표준프레임워크 1.0 버전을 사용한다.