베이스64 (Base64)

- description : 베이스64 (Base64)
- author : 오션
- email : shlim@repia.com
- lastupdate : 2022-05-11 Wed

베이스64 (Base64)

1. 적용위치

Apps.repia.com > R-Utils > Crypto - select box - Base64 선택

2. 적용 파일

Appsportal > CryptoServiceImpl.java

2.1 Base64 적용을 위한 클래스 Import

```
import org.apache.commons.codec.binary.Base64
```

2.1.1 Ref Link

org.apache.commons.codec.binary - Class Base64

3.1 문제

- A (원본 데이터) : 암호화된 수집 데이터(html파일)
- B (디코딩 결과) : Crypto-Base64로 Decryption(복호화)한 결과
- C (인코딩 결과) : B를 Crypto-Base64로 Encryption(암호화)한 결과
- D (사이트 디코딩 결과) : Base64 Decode and Encode에서 A를 복호화한 결과
- E (사이트 인코딩 결과) : Base64 Decode and Encode에서 D를 암호화한 결과

문제 : html 파일에 대해, A와 E는 동일한 결과값을 표시하지만, C가 다른 결과를 표시

Example

Crypto-Base64 암호화 결과

```
입력값 : <head>

encryptInptFld : &lt;head&gt;       // F (입력값이 Crypto 내부에서 변환 & 표시되는 값)

encryptRsltFls : Jmx0O2hlYWQmZ3Q7  // G (F에 Base64가 적용되어 표시되는 결과값)
```

사이트 암호화 결과

```
입력값 : <head>

암호화 결과값 : PGhlYWQ+
```

3.2 해결 방법

3.2.1 기존 코드

```
else if("base64".equals(cryptoVO.getCryptoMethod())) {    // Base64 (binary-to-text encoding schemes)를 선택하고
    if(cryptoVO.getEncryptInptFld()!==null) {             // encryption(암호화)를 실행하면

        byte[] encryptInputByte = cryptoVO.getEncryptInptFld().getBytes("UTF-8");    // encryption 입력값을 charset=UTF-8로 인코딩된 byte 형태로 encryptInputByte에 할당하고
resultData.setEncryptRsltFld(Base64.encodeBase64String(encryptInputByte));
// UTF-8로 인코딩된 encryptInputByte를 Base64 문자열로 반환하여 전달
    }
    else if(cryptoVO.getDecryptInptFld()!==null) {
        String decodeStr = new String(Base64.decodeBase64(cryptoVO.getDecryptInptFld()), "UTF-8");  // 입력된 Base64 문자열을 UTF-8로 디코딩한 값을 String 타입의 decodeStr에 할당
        resultData.setDecryptRsltFld(decodeStr);
    }
}
```

3.2.2 1차 수정 코드

```
else if("base64".equals(cryptoVO.getCryptoMethod())) {
    if(cryptoVO.getEncryptInptFld()!==null) {

        byte[] encryptInputByte =
StringEscapeUtils.unescapeHtml(cryptoVO.getEncryptInptFld()).getBytes("UTF-8
");
        // 1차 수정코드 - "<"가 "&lt;"로 변경되지 않도록
StringEscapeUtils.unescapeHtml로 처리 후 변환되도록 조치
resultData.setEncryptRsltFld(Base64.encodeBase64String(encryptInputByte));
    }
    else if(cryptoVO.getDecryptInptFld()!==null) {
        String decodeStr = new
String(Base64.decodeBase64(cryptoVO.getDecryptInptFld()), "UTF-8");
        resultData.setDecryptRsltFld(decodeStr);
    }
}
```

3.2.3 2차 수정 코드

모든 암호화 방식 선택에 대해 미리 cryptoVo.getEncryptInptFld()의 입력값에 대해 인코딩시 미리 전처리 실시하도록 하기와 같이 코드를 추가하고, 2.3.1 기존 코드를 유지

```
...
/* 인코딩시 전처리 실행 '<' -> '&lt;' -> '<' */

if(StringUtils.isNotEmpty(cryptoVO.getEncryptInptFld())){     // 추가된 코드
    String unEscapeInputField =
StringEscapeUtils.unescapeHtml(cryptoVO.getEncryptInptFld());
    cryptoVO.setEncryptInptFld(unEscapeInputField);

    log.info("[{}]/[{}]/[{}]", cryptoVO.getEncryptInptFld(),
StringEscapeUtils.escapeHtml(cryptoVO.getEncryptInptFld()),
StringEscapeUtils.unescapeHtml(cryptoVO.getEncryptInptFld()));
}
...
...
...
else if("base64".equals(cryptoVO.getCryptoMethod())) {
    if(cryptoVO.getEncryptInptFld()!==null) {

        byte[] encryptInputByte =
cryptoVO.getEncryptInptFld().getBytes("UTF-8");
resultData.setEncryptRsltFld(Base64.encodeBase64String(encryptInputByte));
    }
    else if(cryptoVO.getDecryptInptFld()!==null) {
```

```
        String decodeStr = new
String(Base64.decodeBase64(cryptoVO.getDecryptInptFld()), "UTF-8");
        resultData.setDecryptRsltFld(decodeStr);
    }
}
```

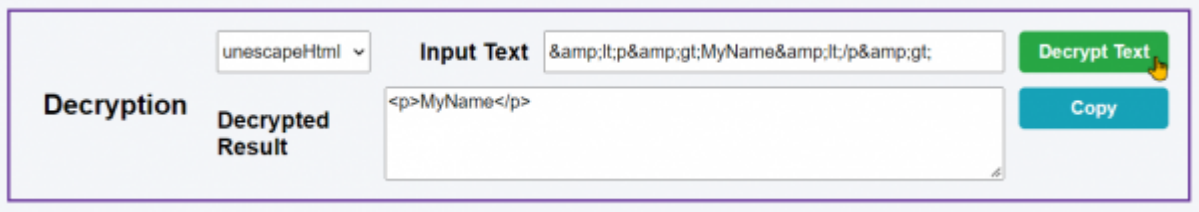
3.2 문제

Crypto-unescapeHtml에서 Decrypt Text 실행 시 에러 발생

Decrypt Text 버튼 클릭 전



Decrypt Text 버튼 클릭 후



결과값이 출력되면서 동시에 input Text란에 입력했던 값이 **<p>MyName</p>** → **<p>MyName</p>**으로 변경됨.

3.2.1 - 코드

코드 수정 전

```
<!-- Input Encrypted ID or Password field -->
<label for="decryptInptText" class="decryptInptTtl">Input Text</label>
<input id="decryptInptText" class="decryptInptFld" name="decryptInptFld"
type="text" placeholder="암호화된 ID 또는 Password를 입력하세요" value="<c:out
value="${inputParam.decryptInptFld}" />"/>
```

코드 수정 후

```
<!-- Input Encrypted ID or Password field -->
<label for="decryptInptText" class="decryptInptTtl">Input Text</label>
<input id="decryptInptText" class="decryptInptFld" name="decryptInptFld"
type="text" placeholder="암호화된 ID 또는 Password를 입력하세요" value="<c:out
```

```
value="${inputParam.decryptInptFld}" escapeXml="false"/>"/>
```

코드 수정 후 Decrypt Text 버튼 클릭 후의 캡쳐



결과값이 출력되면서 동시에 input Text란에 입력했던 값 **<p>MyName</p>**이 변하지 않고 그대로 표시됨을 확인.

Ref Link

[JSTL] Tag가 jsp화면에 그대로 노출될 경우, escapeXml
JSTL Core - Tag out

JSTL Core - Tag out에서 **escapeXml**에 대한 설명은 아래와 같습니다.

> Determines whether characters <,>,&,',“ in the resulting string should be converted to their corresponding character entity codes. Default value is true.
> 결과 문자열의 문자 **<**, **>**, **&**, **'**, **"**를 해당 문자 엔티티 코드로 변환해야 하는지 여부를 결정합니다. 기본값은 true입니다.

오션,, base64

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다.** - 2023.12

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:miscellaneous:base64&rev=1652309995

Last update: **2022/05/12 07:59**