

64 (Base64)

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

Base64

1.

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

2.

Appsportal > CryptoServiceImpl.java

2.1 Base64

Import

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

2.1.1 Ref Link

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

3.1

- A () : (html)
- B () : Crypto-Base64 Decryption()
- C () : B Crypto-Base64 Encryption()
- D () : [Base64 Decode and Encode](#) A
- E () : [Base64 Decode and Encode](#) D

: html , A E , C가

Example

Crypto-Base64

```
 : <head>
encryptInptFld : &lt;head&gt; // F ( Crypto &
)
encryptRsltFls : Jmx002hlyWQmZ3Q7 // G (F Base64가 )
```

```
 : <head>
 : PGhlyWQ+
```

3.2

3.2.1

```
else if("base64".equals(cryptoV0.getCryptoMethod())) { // Base64 (binary-
to-text encoding schemes)
    if(cryptoV0.getEncryptInptFld()!=null) { // encryption(
)

        byte[] encryptInputByte =
cryptoV0.getEncryptInptFld().getBytes("UTF-8"); // encryption
charset=UTF-8 byte encryptInputByte
resultData.setEncryptRsltFld(Base64.encodeBase64String(encryptInputByte));
// UTF-8 encryptInputByte Base64
    }
    else if(cryptoV0.getDecryptInptFld()!=null) {
        String decodeStr = new
String(Base64.decodeBase64(cryptoV0.getDecryptInptFld()), "UTF-8"); //
Base64 UTF-8 String decodeStr
        resultData.setDecryptRsltFld(decodeStr);
    }
}
```

3.2.2 1

```

else if("base64".equals(cryptoV0.getCryptoMethod())) {
    if(cryptoV0.getEncryptInptFld()!=null) {

        byte[] encryptInputByte =
StringEscapeUtils.unescapeHtml(cryptoV0.getEncryptInptFld()).getBytes("UTF-8
");
        // 1 - "<"가 "&lt;"
StringEscapeUtils.unescapeHtml
resultData.setEncryptRsltFld(Base64.encodeBase64String(encryptInputByte));
    }
    else if(cryptoV0.getDecryptInptFld()!=null) {
        String decodeStr = new
String(Base64.decodeBase64(cryptoV0.getDecryptInptFld()), "UTF-8");
        resultData.setDecryptRsltFld(decodeStr);
    }
}
}

```

3.2.3 2

cryptoVo.getEncryptInptFld()
가 , 2.3.1

```

...
/*          '<' -> '&lt;';' -> '<' */

if(StringUtils.isEmpty(cryptoV0.getEncryptInptFld())){ // 가
    String unEscapeInputField =
StringEscapeUtils.unescapeHtml(cryptoV0.getEncryptInptFld());
    cryptoV0.setEncryptInptFld(unEscapeInputField);

    log.info("[{}]/[{}]/[{}]", cryptoV0.getEncryptInptFld(),
StringEscapeUtils.escapeHtml(cryptoV0.getEncryptInptFld()),
StringEscapeUtils.unescapeHtml(cryptoV0.getEncryptInptFld()));
}
...
...
...
else if("base64".equals(cryptoV0.getCryptoMethod())) {
    if(cryptoV0.getEncryptInptFld()!=null) {

        byte[] encryptInputByte =
cryptoV0.getEncryptInptFld().getBytes("UTF-8");
resultData.setEncryptRsltFld(Base64.encodeBase64String(encryptInputByte));
    }
    else if(cryptoV0.getDecryptInptFld()!=null) {

```

```
String decodeStr = new
String(Base64.decodeBase64(cryptoV0.getDecryptInptFld()), "UTF-8");
resultData.setDecryptRsltFld(decodeStr);
}
}
```

3.2

Crypto-unescapeHtml Decrypt Text

Decrypt Text

Decryption

unescapeHtml

Input Text <p>MyName</p>

Decrypted Result

Decrypt Text

Copy

Decrypt Text

Decryption

unescapeHtml

Input Text &lt;p&gt;MyName&lt;/p&gt;

Decrypted Result <p>MyName</p>

Decrypt Text

Copy

input Text <p>MyName</p> →
&lt;p&gt;MyName&lt;/p&gt;

3.2.1 -

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

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

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

Decrypt Text

Decryption	unescapeHtml	Input Text	<p>MyName<p>	Decrypt Text
	Decrypted Result		<p>MyName<p>	Copy

input Text

<p>MyName<p>

Ref Link

[JSTL] Tag가 jsp , escapeXml
JSTL Core - Tag out

JSTL Core - Tag out **escapeXml**

Determines whether characters <, >, &, ' , " in the resulting string should be converted to their corresponding character entity codes. Default value is true.

true <, >, &, ' , "

., base64

From: <http://125.132.25.164/dokuwiki/> -

. - 2023.12

Permanent link: <http://125.132.25.164/dokuwiki/doku.php?id=wiki:miscellaneous:base64&rev=1652262271>

Last update: 2022/05/11 18:44

