



# JavaでURLエンコード/デコードする方法と注意点

投稿者：OSCA

B!ブックマーク 4

Tweet

シェア 0

Pocket

18

[PR] "東野・岡村の旅猿"で登場したロケ地を紹介するファンサイト「あの場所へ行こう！」はこちら。

本稿では Java 言語を用いて文字列をURLエンコード/デコードする方法について解説します。また URL エンコードにおける注意点についても解説します。

## Java で利用できる URL エンコードクラス

まずは Java で利用できる主な URL エンコードクラスを整理します。次の2つから選択すれば良いでしょう。

1. java.net.URLEncoder / URLDecoder (Java標準)
2. Apache Commons Codec の URLCodec クラス

## java.net.URLEncoder / URLDecoder を利用する例

Java 標準の java.net.URLEncoder と java.net.URLDecoder を利用する場合は、次のようにプログラムを書きます。

```
String target = "テストメッセージ";

// エンコードの例
String encodedResult = URLEncoder.encode(target, "UTF-8");
System.out.println("エンコード結果:" + encodedResult);

// デコードの例
String decodedResult = URLDecoder.decode(encodedResult, "UTF-8");
System.out.println("デコード結果:" + decodedResult);
```

実行結果は次の通りです。

```
エンコード結果:%E3%83%86%E3%82%B9%E3%83%88%E3%83%A1%E3%83%83%E3%82%BB%E3%83%BC%E3%82%B8
デコード結果:テストメッセージ
```

## Apache Commons Codec の URLCodec を利用する例

Apache Commons Codec の URLCodec クラスを利用する場合は、次のようにプログラムを書きます。

```
URLCodec codec = new URLCodec("UTF-8");
String target = "テストメッセージ";

// エンコードの例
String encodedResult = codec.encode(target, "UTF-8");
System.out.println("エンコード結果:" + encodedResult);

// デコードの例
String decodedResult = codec.decode(encodedResult, "UTF-8");
System.out.println("デコード結果:" + decodedResult);
```

実行結果は次の通りです。

```
エンコード結果:%E3%83%86%E3%82%B9%E3%83%88%E3%83%A1%E3%83%83%E3%82%BB%E3%83%BC%E3%82%B8
デコード結果:テストメッセージ
```

## どちらを利用するか？

筆者の見解としては、Apache Commons Codec の URLCodec を利用する方が経験上は良いかなと思っています。理由としては次のことがあります。

- Shift\_JIS などの文字コードの URL エンコード/デコードが正しく処理できる。  
(Java 標準の URLEncoder, URLDecoder で Shift\_JIS の文字列が上手く処理できない問題はググるとたくさん出てくる)

## URLエンコードには落とし穴があるぞ！

さて、実はここが本題です。開発者の間で「このパラメータは、URLエンコードしてね」と意思疎通をとったつもりでも、いざ動かしてみると問題が発生する「落とし穴」があります。ここではその「落とし穴」について掘り下げてみようと思います。

URL エンコードにも様々な"方言"があります。方言の違いにより、メッセージが上手く解釈されない場合があります。Java 言語での URLエンコード処理では「\*」「-」「\_」の3つの記号が変換されません。また、半角スペースは「+」に置き換わります。これは不具合という訳ではなく、そういうルール of URLエンコードなのです。URLDecoder についてもその仕様に則ってデコードします。

構築するシステム全体がこのルールに則っていれば何ら問題は発生しませんが、問題が生じやすいのは開発言語をまたいで URL エンコードされた文字列を処理する場合です。例えば JavaScript の URL エンコード処理では、半角スペースは「%20」に置き換わります。また JavaScript の encodeURI 関数では「;/?:@&=+\$\_~\*.,()#」の記号がエンコードされません。フロントサイドに JavaScript を多用する昨今、JavaScript で生成したエンコード文字列を Java で処理する、またその逆の場合においても、エンコード/デコードが上手くいかなくなることは、良く起こりがちな問題です。

環境や言語を踏まえて、設計の段階でどのURLエンコードルールに合わせるのかを明確にしておくことが重要です。