

URI ENCODE / DECODE 에 대해

JAVA

▼ 주로 사용하는 Encoder / Decoder → `java.net.encoder` / `java.net.decoder`

▼ @Entity로 정의하는 객체

▼ 데이터가 변해도 식별자로 지속해서 추적 가능

▼ 예) 회원 엔티티의 키나 나이 값을 변경해도, 식별자로 인식 가능

▼ `java.net.encoder`

- The alphanumeric characters "a" through "z", "A" through "Z" and "0" through "9" remain the same.
- The special chracters . , - , * and _ remain the same.
- The space character " " is converted into a plus sign "+".

▼ `java.net.decoder`

- The alphanumeric characters "a" through "z", "A" through "Z" and "0" through "9" remain the same.
- The special Characters . , - , * and _ remain the same.
- The plus sign "+" is converted into a space character " ".



1. `URLEncoder.encode(String e)` 는 deprecated 되었다. 플랫폼의 디폴트 인코딩 방식을 따르게 되기 때문에, 더욱 명시적인 `URLEncoder.encode(String e, String e[ex "UTF-8"])` 을 권장. decode도 마찬가지

JAVASCRIPT

▼ `encodeURIComponent`

▼ A-Z, a-z, 0-9 ; / ? : @ & = + \$, - _ . ! ~ * ' () # & 에 대해 escape 하지 않는다.

▼ `encodeURIComponent`

▼ A-Z,a-z,0-9 `- _ . ! ~ * ' ()` 에 대해 escape 하지 않는다.

▼ decodeURI

▼ decodeURIComponent

▼ URL 전체를 통째로 인코딩 할 경우 EncodeURI → 디코딩은 DecodeURI

▼ URL parameter나 일부 데이터를 인코딩 할 경우 EncodeURIComponent → 디코딩은 DecodeURIComponent



결론은 + 문자를 쓸 때 주의 해야 한다. 또한 전체 URL을 인코딩 하는 부분이 아니라면, script 단에서는 encodeURIComponent, decodeURIComponent를 사용해야 하며, Java Framework Server 단에서 encoding 된 데이터와 같은 경우, decodeURIComponent("data").replaceAll("+", "%20") 로 + 에 대한 부분을 신경 써줘야 한다. script 단에서 encoding된 데이터와 같은 경우 java Framework Server 단에서 script에서 encodeURIComponent를 썼다는 확신이 없을 시에는 꼭 + 에 대한 방어 처리를 넣어줘야 한다. ex) + → "%20" 로 치환 후 사용

```
public static void main(String[] args) throws UnsupportedEncodingException {  
    System.out.println(URLDecoder.decode(" ; / ? : @ & = %2b $ , - _ . ! ~ * ' ( ) # &", "UTF-8"));  
}
```

Run: RedisSampleApplication x
"/Applications/IntelliJ IDEA.app/Contents/jbr/Contents/Home/bin/java" ...
; / ? : @ & = + \$, - _ . ! ~ * ' () # &