Como codificar a URL em Java ME

Este artigo explica como codificar uma URL em Java ME.

```
// O seguinte método é usado para codificar a String da URL,

// o método retorna a String da URL codificada

public static String urlEncode(String s) {
    StringBuffer sbuf = new StringBuffer();
    int len = s.length();
    for (int i = 0; i < len; i++) {
        int ch = s.charAt(i);
        if ('A' <= ch && ch <= 'Z') { // 'A'..'Z'
            sbuf.append((char)ch);
        } else if ('a' <= ch && ch <= 'z') { // 'a'..'z'
            sbuf.append((char)ch);
        } else if ('0' <= ch && ch <= '9') { // '0'..'9'
            sbuf.append((char)ch);
        } else if (ch == ' ') { // space
            sbuf.append('+');
        } else if (ch == '-' || ch == '_' // Sem reservas
                || ch == '.' || ch == '!'
                || ch == '~' || ch == '*'
                || ch == '\\' || ch == '('
                || ch == ')' || ch == '/'
         || ch == '&' || ch == '='
         || ch == '?') {
            sbuf.append((char)ch);
        } else if (ch <= 0x007f) { // ou código ASCII
            sbuf.append(hex[ch]);
        } else if (ch <= 0x07FF) { // não-ASCII <= 0x7FF
            sbuf.append(hex[0xc0 | (ch >> 6)]);
            sbuf.append(hex[0x80 | (ch & 0x3F)]);
        } else { // 0x7FF < ch <= 0xFFFF
            sbuf.append(hex[0xe0 | (ch >> 12)]);
            sbuf.append(hex[0x80 | ((ch >> 6) & 0x3F)]);
            sbuf.append(hex[0x80 | (ch & 0x3F)]);
        }
    }
    return sbuf.toString();
}


// Constantes Hex.
final static String[] hex = {
    "%00", "%01", "%02", "%03", "%04", "%05", "%06", "%07",
    "%08", "%09", "%0a", "%0b", "%0c", "%0d", "%0e", "%0f",
    "%10", "%11", "%12", "%13", "%14", "%15", "%16", "%17",
    "%18", "%19", "%1a", "%1b", "%1c", "%1d", "%1e", "%1f",
    "%20", "%21", "%22", "%23", "%24", "%25", "%26", "%27",
    "%28", "%29", "%2a", "%2b", "%2c", "%2d", "%2e", "%2f",
    "%30", "%31", "%32", "%33", "%34", "%35", "%36", "%37",
    "%38", "%39", "%3a", "%3b", "%3c", "%3d", "%3e", "%3f",
    "%40", "%41", "%42", "%43", "%44", "%45", "%46", "%47",
```


(C) Copyright Nokia 2014. All rights reserved.

```
    "%48", "%49", "%4a", "%4b", "%4c", "%4d", "%4e", "%4f",
    "%50", "%51", "%52", "%53", "%54", "%55", "%56", "%57",
    "%58", "%59", "%5a", "%5b", "%5c", "%5d", "%5e", "%5f",
    "%60", "%61", "%62", "%63", "%64", "%65", "%66", "%67",
    "%68", "%69", "%6a", "%6b", "%6c", "%6d", "%6e", "%6f",
    "%70", "%71", "%72", "%73", "%74", "%75", "%76", "%77",
    "%78", "%79", "%7a", "%7b", "%7c", "%7d", "%7e", "%7f",
    "%80", "%81", "%82", "%83", "%84", "%85", "%86", "%87",
    "%88", "%89", "%8a", "%8b", "%8c", "%8d", "%8e", "%8f",
    "%90", "%91", "%92", "%93", "%94", "%95", "%96", "%97",
    "%98", "%99", "%9a", "%9b", "%9c", "%9d", "%9e", "%9f",
    "%a0", "%a1", "%a2", "%a3", "%a4", "%a5", "%a6", "%a7",
    "%a8", "%a9", "%aa", "%ab", "%ac", "%ad", "%ae", "%af",
    "%b0", "%b1", "%b2", "%b3", "%b4", "%b5", "%b6", "%b7",
    "%b8", "%b9", "%ba", "%bb", "%bc", "%bd", "%be", "%bf",
    "%c0", "%c1", "%c2", "%c3", "%c4", "%c5", "%c6", "%c7",
    "%c8", "%c9", "%ca", "%cb", "%cc", "%cd", "%ce", "%cf",
    "%d0", "%d1", "%d2", "%d3", "%d4", "%d5", "%d6", "%d7",
    "%d8", "%d9", "%da", "%db", "%dc", "%dd", "%de", "%df",
    "%e0", "%e1", "%e2", "%e3", "%e4", "%e5", "%e6", "%e7",
    "%e8", "%e9", "%ea", "%eb", "%ec", "%ed", "%ee", "%ef",
    "%f0", "%f1", "%f2", "%f3", "%f4", "%f5", "%f6", "%f7",
    "%f8", "%f9", "%fa", "%fb", "%fc", "%fd", "%fe", "%ff"
};
```

As codificações são usadas para maior segurança quanto restringir o acesso aos usuários mais experientes fazendo uso de por exemplo uma espécie de *Code injection* (Injeção de código) na URL para acessar páginas não autorizadas.

(C) Copyright Nokia 2014. All rights reserved.