

## 24.6.7 URL

Die Klasse `URL` (`gb.web`) stellt vier (statische) Methoden zum Umgang mit URL-Strings zur Verfügung. Ein `URL` (Uniform Resource Locator) ist ein Bezeichner für eine Netzwerk-Ressource.

### 24.6.7.1 Methoden

Die Klasse `URL` verfügt nur über diese vier Methoden:

Methoden	Rückgabotyp	Beschreibung
<code>Encode ( Path As String )</code>	<code>String</code>	Kodiert einen URL in das <code>Application/x-www-form-urlencoded</code> -Format.
<code>Decode ( Path As String )</code>	<code>String</code>	Dekodiert einen URL aus dem <code>Application/x-www-form-urlencoded</code> -Format.
<code>SetQuery ( URL As String, Field As String, Value As String )</code>	<code>String</code>	Setzt den Wert des angegebenen Feldes in einer Abfrage und gibt die modifizierte Abfrage zurück. Wenn das angegebene Feld nicht vorhanden ist, so wird es zur Abfrage hinzugefügt.
<code>UnsetQuery ( URL As String, Field As String )</code>	<code>String</code>	Entfernt das angegebene Feld aus dem Abfrage-URL und gibt die modifizierte Abfrage zurück. Hat der URL keinen Feld-Parameter, dann wird die unveränderte Abfrage zurückgegeben.

Tabelle 24.6.7.1.1 : Methoden der Klasse `URL`

Die beiden in der Tabelle 24.6.7.1.1 aufgeführten Methoden `URL.Encode` und `URL.Decode` aus der Klasse `URL` entsprechen den gleichnamigen Methoden der Klasse `CGI` (`gb.web`), die im Kapitel 24.6.1 `CGI` beschrieben werden.

### 24.6.7.2 Beispiele

Die folgenden Beispiele nutzen die vier Methoden der Klasse `URL`.

Die angegebenen URLs werden zuerst in das `Application/x-www-form-urlencoded`-Format kodiert und dann wieder de-kodiert:

```
Print URL.Encode("http://www.gambas-buch.de/dw/doku.php?id=k18:start")
Print URL.Decode("http%3A%2F%2Fwww.gambas-buch.de%2Fdw%2Fdoku.php%3Fid%3Dk18%3Astart")

http%3A%2F%2Fwww.gambas-buch.de%2Fdw%2Fdoku.php%3Fid%3Dk18%3Astart
http://www.gambas-buch.de/dw/doku.php?id=k18:start
```

Diese Beispiele demonstrieren die Änderung einer bestehenden HTTP-Anfrage:

```
Print SetQuery("http://example.com/test", "a", "1")      ' Das Feld `a` bekommt den Wert 1
http://example.com/test?a=1

Print SetQuery("http://example.com/test?a=1", "b", "2")  ' Das Feld `b` wird hinzugefügt mit dem Wert 2
http://example.com/test?a=1&b=2

Print SetQuery("http://example.com/test?a=1&b=2", "a", "3") ' Nur das Feld `a` bekommt den Wert 3
http://example.com/test?b=2&a=3

Print SetQuery("http://example.com/test?b=2&a=3", "b", "") ' Nur das Feld `b` bekommt einen Leerstring
http://example.com/test?a=3&b
```

Bei den nächsten Aufrufen der Methode `UnsetQuery(...)` werden Felder aus einer Anfrage entfernt. Aber nur dann, wenn das angegebene Feld im Original der Anfrage existiert:

```
Print UnsetQuery("http://example.com/test?a=3&b", "a") ' Das Feld `a` wird aus der Anfrage entfernt
http://example.com/test?b

Print UnsetQuery("http://example.com/test?b", "a")     ' Das Feld `a` wird aus der Anfrage entfernt
http://example.com/test?b

Print UnsetQuery("http://example.com/test?b", "b")     ' Das Feld `b` wird aus der Anfrage entfernt
http://example.com/test

Print UnsetQuery("http://example.com/test?a=3&b")     ' Die Anfrage wird im Original zurückgegeben
http://example.com/test? a=3&b
```