



ZINCKSOFT

**Give The Dream New Life.**

*JSU.URL*

*The complete Module API Reference*



<b>Quick Introduction</b>	<b>1</b>
Including the module in your page	1
<b>JSU.URL API Reference</b>	<b>2</b>
<b>Parsing Methods</b>	<b>2</b>
<i>getQueryString</i>	2
<i>getArguments</i>	2
<i>getProtocol</i>	3
<i>getTagName</i>	3
<i>getFileName</i>	4
<i>getFileType</i>	4
<i>getFromTag</i>	5
<b>Validation Methods</b>	<b>5</b>
<i>isValidArgument</i>	5
<i>isValidHash</i>	6
<i>isValidQueryString</i>	6
<i>isValid</i>	7
<i>isValidFriendly</i>	7
<i>isValidHTMLTag</i>	8

# Quick Introduction

The JSU URL Module (**JSU.URL** / **JSUURL**) provides various methods for working with URL's and its components, but also some validation methods not found in the validation module for being URL-specific.

## Including the module in your page

Before including this module in your page, first include the core module:

```
<script type="text/javascript" src="jsu.core.js"></script>  
<script type="text/javascript" src="jsu.url.js"></script>
```

# JSU.URL API Reference

## A. Parsing Methods

### 1. getQueryString

Obtains the query string from an URL address.

**Syntax:**

```
getQueryString(url)
```

**Arguments:**

- **url**: the url to parse.

**Returns:** A string containing the URL's query string.

**Example:**

```
alert(JSU.URL.getQueryString("www.x.it/a.php?test=2&p=3")); // test=2&p=3
alert(JSU.URL.getQueryString("a.php?test=%20&p=3#top")); // test=%20&p=3#top
alert(JSU.URL.getQueryString("a.php?")); // empty string
```

### 2. getArguments

Parses an URL's query string returning all the (key, value) pairs.

**Syntax:**

```
getArguments(url [, separator [, justArgs]])
getArguments(url [, justArgs])
```

**Arguments:**

- **url**: the url to parse;
- **separator**: key/value pairs separator; defaults to "&";
- **justArgs**: a boolean indicating whether the url is a full url or just the query string; defaults to false, so by default you have to specify a full URL for the first argument.

**Returns:** An array containing all the key/value pairs as strings. If the query string has a hash (also called *fragment*), the last element in the array will be that hash value.

**Example:**

```

alert(JSU.URL.getArguments("www.x.it/a.php?test=2&p=3"));
// ["test=2", "p=3"]
alert(JSU.URL.getArguments("a.php?test=%20&p=3#top"));
// ["test=%20", "p=3", "#top"]
alert(JSU.URL.getArguments("a.php?"));
// [] - empty array
alert(JSU.URL.getArguments("a.php?test=2;a=3", ";"));
// ["test=2", "a=3"]
alert(JSU.URL.getArguments("test=3&a=2;b=3#hash", true));
// ["test=3", "a=2;b=2", "#hash"]

```

**3. getProtocol**

Obtains an URL's protocol.

**Syntax:**

```
getProtocol(url)
```

**Arguments:**

- **url**: the url to parse.

**Returns:** A string containing the URL's protocol.

**Example:**

```

alert(JSU.URL.getProtocol("http://www.x.com")); // http
alert(JSU.URL.getProtocol("ftp://121.4.18.13")); // ftp
alert(JSU.URL.getProtocol("www.test.com")); // empty string

```

**4. getTagName**

Obtains the tag name from an HTML tag.

**Syntax:**

```
getTagName(tag)
```

**Arguments:**

- **tag**: the HTML tag.

**Returns:** A string containing the tag name.

**Example:**

```
alert(JSU.URL.getTagName("<br/>")); // br
alert(JSU.URL.getTagName("<a href='src'></a>")); // a
alert(JSU.URL.getTagName("<html>")); // html
```

**5. getFileName**

Obtains the filename from a specified path or URL. A valid filename contains a name and an extension separated by a dot.

**Syntax:**

```
getFileName(s)
```

**Arguments:**

- *s*: the path or URL to parse.

**Returns:** A string containing the filename.

**Example:**

```
alert(JSU.URL.getFileName("www.test.com/a.php")); // a.php
alert(JSU.URL.getFileName("www.test.com/a.php?test=2")); // a.php
alert(JSU.URL.getFileName("C:\\test\\file.inc")); // file.inc
```

**6. getFileType**

Obtains the extension from a filename component of a path or URL.

**Syntax:**

```
getFileType(s)
```

**Arguments:**

- *s*: the path or URL to parse.

**Returns:** A string containing the filename.

**Example:**

```
alert(JSU.URL.getFileType("www.test.com/a.php")); // php
alert(JSU.URL.getFileType("www.test.com/a.php?test=2")); // php
alert(JSU.URL.getFileType("C:\\test\\file.inc")); // inc
```

## 7. getFromTag

Parses an HTML tag and obtains its URL component, whether it's from an href attribute, an action attribute, and so on. Supports all HTML tags that contain an attribute with an URL value (**a**, **link**, **base**, **form**, **img**, **embed**, **bgsound**, **input**, **script**, **frame**, **iframe**, **area**, **head**, **html**, **ins**, **q**, **object**).

### Syntax:

```
getFromTag(tag)
```

### Arguments:

- **tag**: the HTML tag to parse.

**Returns:** A string containing the tag's URL.

### Example:

```
alert(JSU.URL.getFromTag("<a href='test.inc'/>")); // test.inc
alert(JSU.URL.getFromTag("<form action=\"file.php\"></form>")); // file.php
alert(JSU.URL.getFromTag("<img src='test.jpg' />")); // test.jpg
alert(JSU.URL.getFromTag("<iframe src='f.html'>")); // f.html
alert(JSU.URL.getFromTag("<head profile='prf'>")); // prf
alert(JSU.URL.getFromTag("<html xmlns='a.b'>")); // a.b
alert(JSU.URL.getFromTag("<q cite='file'>")); // file
alert(JSU.URL.getFromTag("<object data='test.x'>")); // test.x
alert(JSU.URL.getFromTag("<object codebase='test.x'>")); // test.x
```

## B. Validation Methods

### 1. isValidArgument

Checks if a URL query string's argument is valid.

### Syntax:

```
isValidArgument(qsArg)
```

### Arguments:

- **qsArg**: the query string's argument to parse.

**Returns:** A boolean.



**Example:**

```
alert(JSU.URL.isValidArgument("test=3")); // true
alert(JSU.URL.isValidArgument("a")); // true
alert(JSU.URL.isValidArgument("a='2'%20")); // true
alert(JSU.URL.isValidArgument("a=3;b=4")); // false
```

**2. isValidHash**

Checks if a URL query string's hash is valid.

**Syntax:**

```
isValidHash(qsHash)
```

**Arguments:**

- **qsHash**: the query string's hash to parse - usually obtained from the last element of an array returned by the **getArguments** method.

**Returns:** A boolean.

**Example:**

```
alert(JSU.URL.isValidHash("test")); // false
alert(JSU.URL.isValidHash("#top")); // true
```

**3. isValidQueryString**

Checks if an URL's query string is valid by checking all its arguments, including the hash.

**Syntax:**

```
isValidQueryString(qs [, separator])
```

**Arguments:**

- **qs**: the URL's query string;
- **separator**: the query string's arguments separator; defaults to "&".

**Returns:** A boolean.

**Example:**

```

alert(JSU.URL.isValidQueryString("test=2&a=3,4;5,6")); // true
alert(JSU.URL.isValidQueryString("test=2&a=3;b=4")); // false
alert(JSU.URL.isValidQueryString("test=2&a=&c")); // true
alert(JSU.URL.isValidQueryString("test=2&a=&#hash")); // false
alert(JSU.URL.isValidQueryString("test=2&a=&name#hash")); // true

```

**4. isValid**

Checks if an URL is valid by checking all its components (the protocol, the url address and the query string). The URL can be a standard web address or a mail address. If the URL is an email address, that address will be also validated.

**Syntax:**

```
isValid(url)
```

**Arguments:**

- **qs**: the URL's query string.

**Returns:** A boolean.

**Example:**

```

alert(JSU.URL.isValid("http://127.0.0.1")); // true
alert(JSU.URL.isValid("www.test.com")); // true
alert(JSU.URL.isValid("x?:/a.com")); // false
alert(JSU.URL.isValid("www.a.com?a=2#hash")); // true
alert(JSU.URL.isValid("mailto:sender@yahoo.com")); // true
alert(JSU.URL.isValid("mailto:a@y.it?subject=testing")); // true
alert(JSU.URL.isValid("mailto:sender@x")); // false

```

**5. isValidFriendly**

Checks if a string is qualified as a friendly URL component.

**Syntax:**

```
isValidFriendly(s)
```

**Arguments:**

- **s**: a string.

**Returns:** A boolean.

**Example:**

```
alert(JSU.URL.isValidFriendly("a-friendly-name")); // true
alert(JSU.URL.isValidFriendly("not.friendly")); // false
```

**6. isValidHTMLTag**

Checks if an HTML tag is valid. Notice that this method validates only XHTML syntax.

**Syntax:**

```
isValidHTMLTag(tag [, expectedTagName])
```

**Arguments:**

- **tag**: an HTML tag;
- **expectedTagName**: checks if the tag is valid and is also the expected tag.

**Returns:** A boolean.

**Example:**

```
alert(JSU.URL.isValidHTMLTag("<a href='test'></a>")); // true
alert(JSU.URL.isValidHTMLTag("<a href='test'></a>", "a")); // true
alert(JSU.URL.isValidHTMLTag("<br />")); // true
alert(JSU.URL.isValidHTMLTag("<html><body>")); // false
alert(JSU.URL.isValidHTMLTag("<a href='test' />")); // true
alert(JSU.URL.isValidHTMLTag("<img src='x.jpg' width=\"9\"></img>")); // true
alert(JSU.URL.isValidHTMLTag("<br>")); // false - not XHTML
alert(JSU.URL.isValidHTMLTag("<a></a>", "br")); // false
alert(JSU.URL.isValidHTMLTag("<ins cite=\"test'>")); // false
```