

javax.servlet

Interface ServletContext

public interface **ServletContext**

Defines a set of methods that a servlet uses to communicate with its servlet container, for example, to get the MIME type of a file, dispatch requests, or write to a log file.

There is one context per "web application" per Java Virtual Machine. (A "web application" is a collection of servlets and content installed under a specific subset of the server's URL namespace such as `/catalog` and possibly installed via a `.war` file.)

In the case of a web application marked "distributed" in its deployment descriptor, there will be one context instance for each virtual machine. In this situation, the context cannot be used as a location to share global information (because the information won't be truly global). Use an external resource like a database instead.

The `ServletContext` object is contained within the [ServletConfig](#) object, which the Web server provides the servlet when the servlet is initialized.

**Version:**  
\$Version\$

**Author:**  
Various

**See Also:**  
[Servlet.getServletConfig\(\)](#), [ServletConfig.getServletContext\(\)](#)

Method Summary	
java.lang.Object	<a href="#">getAttribute</a> (java.lang.String name) Returns the servlet container attribute with the given name, or <code>null</code> if there is no attribute by that name.
java.util.Enumeration	<a href="#">getAttributeNames</a> () Returns an <code>Enumeration</code> containing the attribute names available within this servlet context.
<a href="#">ServletContext</a>	<a href="#">getContext</a> (java.lang.String uripath) Returns a <code>ServletContext</code> object that corresponds to a specified URL on the server.
java.lang.String	<a href="#">getContextPath</a> ()

	Returns the context path of the web application.
java.lang.String	<a href="#">getInitParameter</a> (java.lang.String name) Returns a String containing the value of the named context-wide initialization parameter, or null if the parameter does not exist.
java.util.Enumeration	<a href="#">getInitParameterNames</a> () Returns the names of the context's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the context has no initialization parameters.
int	<a href="#">getMajorVersion</a> () Returns the major version of the Java Servlet API that this servlet container supports.
java.lang.String	<a href="#">getMimeType</a> (java.lang.String file) Returns the MIME type of the specified file, or null if the MIME type is not known.
int	<a href="#">getMinorVersion</a> () Returns the minor version of the Servlet API that this servlet container supports.
<a href="#">RequestDispatcher</a>	<a href="#">getNamedDispatcher</a> (java.lang.String name) Returns a <a href="#">RequestDispatcher</a> object that acts as a wrapper for the named servlet.
java.lang.String	<a href="#">getRealPath</a> (java.lang.String path) Returns a String containing the real path for a given virtual path.
<a href="#">RequestDispatcher</a>	<a href="#">getRequestDispatcher</a> (java.lang.String path) Returns a <a href="#">RequestDispatcher</a> object that acts as a wrapper for the resource located at the given path.
java.net.URL	<a href="#">getResource</a> (java.lang.String path) Returns a URL to the resource that is mapped to a specified path.
java.io.InputStream	<a href="#">getResourceAsStream</a> (java.lang.String path) Returns the resource located at the named path as an InputStream object.
java.util.Set	<a href="#">getResourcePaths</a> (java.lang.String path) Returns a directory-like listing of all the paths to resources within the web application whose longest sub-path matches the supplied path argument.
java.lang.String	<a href="#">getServerInfo</a> () Returns the name and version of the servlet container on which the servlet is running.
<a href="#">Servlet</a>	<a href="#">getServlet</a> (java.lang.String name) <b>Deprecated.</b> <i>As of Java Servlet API 2.1, with no direct replacement.</i>  <i>This method was originally defined to retrieve a servlet from a ServletContext. In this version, this method always returns null and</i>

	<p><i>remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.</i></p> <p><i>In lieu of this method, servlets can share information using the <code>ServletContext</code> class and can perform shared business logic by invoking methods on common non-servlet classes.</i></p>
<code>java.lang.String</code>	<p><a href="#"><code>getServletContextName()</code></a></p> <p>Returns the name of this web application corresponding to this <code>ServletContext</code> as specified in the deployment descriptor for this web application by the display-name element.</p>
<code>java.util.Enumeration</code>	<p><a href="#"><code>getServletNames()</code></a></p> <p><b>Deprecated.</b> <i>As of Java Servlet API 2.1, with no replacement.</i></p> <p><i>This method was originally defined to return an <code>Enumeration</code> of all the servlet names known to this context. In this version, this method always returns an empty <code>Enumeration</code> and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.</i></p>
<code>java.util.Enumeration</code>	<p><a href="#"><code>getServlets()</code></a></p> <p><b>Deprecated.</b> <i>As of Java Servlet API 2.0, with no replacement.</i></p> <p><i>This method was originally defined to return an <code>Enumeration</code> of all the servlets known to this servlet context. In this version, this method always returns an empty enumeration and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.</i></p>
<code>void</code>	<p><a href="#"><code>log</code></a>(<code>java.lang.Exception exception</code>, <code>java.lang.String msg</code>)</p> <p><b>Deprecated.</b> <i>As of Java Servlet API 2.1, use <a href="#"><code>log(String message, Throwable throwable)</code></a> instead.</i></p> <p><i>This method was originally defined to write an exception's stack trace and an explanatory error message to the servlet log file.</i></p>
<code>void</code>	<p><a href="#"><code>log</code></a>(<code>java.lang.String msg</code>)</p> <p>Writes the specified message to a servlet log file, usually an event log.</p>
<code>void</code>	<p><a href="#"><code>log</code></a>(<code>java.lang.String message</code>, <code>java.lang.Throwable throwable</code>)</p> <p>Writes an explanatory message and a stack trace for a given <code>Throwable</code> exception to the servlet log file.</p>
<code>void</code>	<p><a href="#"><code>removeAttribute</code></a>(<code>java.lang.String name</code>)</p> <p>Removes the attribute with the given name from the servlet context.</p>
<code>void</code>	<p><a href="#"><code>setAttribute</code></a>(<code>java.lang.String name</code>, <code>java.lang.Object object</code>)</p>

	Binds an object to a given attribute name in this servlet context.
--	--

## Method Detail

### getContextPath

```
java.lang.String getContextPath()
```

Returns the context path of the web application.

The context path is the portion of the request URI that is used to select the context of the request. The context path always comes first in a request URI. The path starts with a "/" character but does not end with a "/" character. For servlets in the default (root) context, this method returns "".

It is possible that a servlet container may match a context by more than one context path. In such cases the [HttpServletRequest.getContextPath\(\)](#) will return the actual context path used by the request and it may differ from the path returned by this method. The context path returned by this method should be considered as the prime or preferred context path of the application.

**Since:**

Servlet 2.5

**See Also:**

[HttpServletRequest.getContextPath\(\)](#)

---

### getContext

```
ServletContext getContext(java.lang.String uripath)
```

Returns a `ServletContext` object that corresponds to a specified URL on the server.

This method allows servlets to gain access to the context for various parts of the server, and as needed obtain [RequestDispatcher](#) objects from the context. The given path must begin with "/", is interpreted relative to the server's document root and is matched against the context roots of other web applications hosted on this container.

In a security conscious environment, the servlet container may return `null` for a given URL.

**Parameters:**

`uripath` - a `String` specifying the context path of another web application in the container.

**Returns:**

the `ServletContext` object that corresponds to the named URL, or `null` if either none exists or the container wishes to restrict this access.

**See Also:**

[RequestDispatcher](#)

## getMajorVersion

```
int getMajorVersion()
```

Returns the major version of the Java Servlet API that this servlet container supports. All implementations that comply with Version 2.4 must have this method return the integer 2.

**Returns:**

2

---

## getMinorVersion

```
int getMinorVersion()
```

Returns the minor version of the Servlet API that this servlet container supports. All implementations that comply with Version 2.4 must have this method return the integer 4.

**Returns:**

4

---

## getMimeType

```
java.lang.String getMimeType(java.lang.String file)
```

Returns the MIME type of the specified file, or `null` if the MIME type is not known. The MIME type is determined by the configuration of the servlet container, and may be specified in a web application deployment descriptor. Common MIME types are `"text/html"` and `"image/gif"`.

**Parameters:**

`file` - a `String` specifying the name of a file

**Returns:**

a `String` specifying the file's MIME type

---

## getResourcePaths

```
java.util.Set getResourcePaths(java.lang.String path)
```

Returns a directory-like listing of all the paths to resources within the web application whose longest sub-path matches the supplied path argument. Paths indicating subdirectory paths end with a `'/'`. The returned paths are all relative to the root of the web application and have a leading `'/'`. For example, for a web application containing

```
/welcome.html  
/catalog/index.html  
/catalog/products.html
```

```
/catalog/offers/books.html  
/catalog/offers/music.html  
/customer/login.jsp  
/WEB-INF/web.xml  
/WEB-INF/classes/com.acme.OrderServlet.class,
```

`getResourcePaths("/")` returns `={"/welcome.html", "/catalog/", "/customer/", "/WEB-INF/"}`

`getResourcePaths("/catalog/")` returns `={"/catalog/index.html", "/catalog/products.html", "/catalog/offers/"}`.

**Parameters:**

`path` - the partial path used to match the resources, which must start with a `/`

**Returns:**

a Set containing the directory listing, or null if there are no resources in the web application whose path begins with the supplied path.

**Since:**

Servlet 2.3

---

## getResource

```
java.net.URL getResource(java.lang.String path)  
throws java.net.MalformedURLException
```

Returns a URL to the resource that is mapped to a specified path. The path must begin with a `/` and is interpreted as relative to the current context root.

This method allows the servlet container to make a resource available to servlets from any source. Resources can be located on a local or remote file system, in a database, or in a `.war` file.

The servlet container must implement the URL handlers and `URLConnection` objects that are necessary to access the resource.

This method returns `null` if no resource is mapped to the pathname.

Some containers may allow writing to the URL returned by this method using the methods of the URL class.

The resource content is returned directly, so be aware that requesting a `.jsp` page returns the JSP source code. Use a `RequestDispatcher` instead to include results of an execution.

This method has a different purpose than `java.lang.Class.getResource`, which looks up resources based on a class loader. This method does not use class loaders.

**Parameters:**

`path` - a `String` specifying the path to the resource

**Returns:**

the resource located at the named path, or `null` if there is no resource at that path

**Throws:**

`java.net.MalformedURLException` - if the pathname is not given in the correct form

---

## getResourceAsStream

`java.io.InputStream` **getResourceAsStream**(`java.lang.String` path)

Returns the resource located at the named path as an `InputStream` object.

The data in the `InputStream` can be of any type or length. The path must be specified according to the rules given in `getResource`. This method returns `null` if no resource exists at the specified path.

Meta-information such as content length and content type that is available via `getResource` method is lost when using this method.

The servlet container must implement the URL handlers and `URLConnection` objects necessary to access the resource.

This method is different from `java.lang.Class.getResourceAsStream`, which uses a class loader. This method allows servlet containers to make a resource available to a servlet from any location, without using a class loader.

### Parameters:

path - a `String` specifying the path to the resource

### Returns:

the `InputStream` returned to the servlet, or `null` if no resource exists at the specified path

---

## getRequestDispatcher

[RequestDispatcher](#) **getRequestDispatcher**(`java.lang.String` path)

Returns a [RequestDispatcher](#) object that acts as a wrapper for the resource located at the given path. A `RequestDispatcher` object can be used to forward a request to the resource or to include the resource in a response. The resource can be dynamic or static.

The pathname must begin with a "/" and is interpreted as relative to the current context root. Use `getContext` to obtain a `RequestDispatcher` for resources in foreign contexts. This method returns `null` if the `ServletContext` cannot return a `RequestDispatcher`.

### Parameters:

path - a `String` specifying the pathname to the resource

### Returns:

a `RequestDispatcher` object that acts as a wrapper for the resource at the specified path, or `null` if the `ServletContext` cannot return a `RequestDispatcher`

### See Also:

[RequestDispatcher](#), [getContext\(java.lang.String\)](#)

---

## getNamedDispatcher

[RequestDispatcher](#) **getNamedDispatcher**(java.lang.String name)

Returns a [RequestDispatcher](#) object that acts as a wrapper for the named servlet.

Servlets (and JSP pages also) may be given names via server administration or via a web application deployment descriptor. A servlet instance can determine its name using

[ServletConfig.getServletName\(\)](#).

This method returns `null` if the `ServletContext` cannot return a `RequestDispatcher` for any reason.

### Parameters:

name - a `String` specifying the name of a servlet to wrap

### Returns:

a `RequestDispatcher` object that acts as a wrapper for the named servlet, or `null` if the `ServletContext` cannot return a `RequestDispatcher`

### See Also:

[RequestDispatcher](#), [getContext\(java.lang.String\)](#),  
[ServletConfig.getServletName\(\)](#)

---

## getServlet

[Servlet](#) **getServlet**(java.lang.String name)  
throws [ServletException](#)

**Deprecated.** *As of Java Servlet API 2.1, with no direct replacement.*

*This method was originally defined to retrieve a servlet from a `ServletContext`. In this version, this method always returns `null` and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.*

*In lieu of this method, servlets can share information using the `ServletContext` class and can perform shared business logic by invoking methods on common non-servlet classes.*

### Throws:

[ServletException](#)

---

## getServlets

java.util.Enumeration **getServlets**()

**Deprecated.** *As of Java Servlet API 2.0, with no replacement.*

*This method was originally defined to return an `Enumeration` of all the servlets known to this servlet context. In this version, this method always returns an empty enumeration and remains only to preserve binary compatibility. This method will be permanently removed in a future version of*



*the Java Servlet API.*

---

## getServletNames

```
java.util.Enumeration getServletNames()
```

**Deprecated.** *As of Java Servlet API 2.1, with no replacement.*

*This method was originally defined to return an `Enumeration` of all the servlet names known to this context. In this version, this method always returns an empty `Enumeration` and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.*

---

## log

```
void log(java.lang.String msg)
```

Writes the specified message to a servlet log file, usually an event log. The name and type of the servlet log file is specific to the servlet container.

**Parameters:**

`msg` - a `String` specifying the message to be written to the log file

---

## log

```
void log(java.lang.Exception exception,  
         java.lang.String msg)
```

**Deprecated.** *As of Java Servlet API 2.1, use [log\(String message, Throwable throwable\)](#) instead.*

*This method was originally defined to write an exception's stack trace and an explanatory error message to the servlet log file.*

---

## log

```
void log(java.lang.String message,  
         java.lang.Throwable throwable)
```

Writes an explanatory message and a stack trace for a given `Throwable` exception to the servlet log file. The name and type of the servlet log file is specific to the servlet container, usually an event log.

**Parameters:**

`message` - a `String` that describes the error or exception

throwable - the Throwable error or exception

---

## getRealPath

```
java.lang.String getRealPath(java.lang.String path)
```

Returns a `String` containing the real path for a given virtual path. For example, the path `"/index.html"` returns the absolute file path on the server's filesystem would be served by a request for `"http://host/contextPath/index.html"`, where `contextPath` is the context path of this `ServletContext`.

The real path returned will be in a form appropriate to the computer and operating system on which the servlet container is running, including the proper path separators. This method returns `null` if the servlet container cannot translate the virtual path to a real path for any reason (such as when the content is being made available from a `.war` archive).

### Parameters:

`path` - a `String` specifying a virtual path

### Returns:

a `String` specifying the real path, or `null` if the translation cannot be performed

---

## getServerInfo

```
java.lang.String getServerInfo()
```

Returns the name and version of the servlet container on which the servlet is running.

The form of the returned string is *servername/versionnumber*. For example, the JavaServer Web Development Kit may return the string `JavaServer Web Dev Kit/1.0`.

The servlet container may return other optional information after the primary string in parentheses, for example, `JavaServer Web Dev Kit/1.0 (JDK 1.1.6; Windows NT 4.0 x86)`.

### Returns:

a `String` containing at least the servlet container name and version number

---

## getInitParameter

```
java.lang.String getInitParameter(java.lang.String name)
```

Returns a `String` containing the value of the named context-wide initialization parameter, or `null` if the parameter does not exist.

This method can make available configuration information useful to an entire "web application". For example, it can provide a webmaster's email address or the name of a system that holds critical data.

**Parameters:**

name - a `String` containing the name of the parameter whose value is requested

**Returns:**

a `String` containing at least the servlet container name and version number

**See Also:**

[`ServletConfig.getInitParameter\(java.lang.String\)`](#)

---

## **getInitParameterNames**

```
java.util.Enumeration getInitParameterNames()
```

Returns the names of the context's initialization parameters as an `Enumeration` of `String` objects, or an empty `Enumeration` if the context has no initialization parameters.

**Returns:**

an `Enumeration` of `String` objects containing the names of the context's initialization parameters

**See Also:**

[`ServletConfig.getInitParameter\(java.lang.String\)`](#)

---

## **getAttribute**

```
java.lang.Object getAttribute(java.lang.String name)
```

Returns the servlet container attribute with the given name, or `null` if there is no attribute by that name. An attribute allows a servlet container to give the servlet additional information not already provided by this interface. See your server documentation for information about its attributes. A list of supported attributes can be retrieved using `getAttributeNames`.

The attribute is returned as a `java.lang.Object` or some subclass. Attribute names should follow the same convention as package names. The Java Servlet API specification reserves names matching `java.*`, `javax.*`, and `sun.*`.

**Parameters:**

name - a `String` specifying the name of the attribute

**Returns:**

an `Object` containing the value of the attribute, or `null` if no attribute exists matching the given name

**See Also:**

[`getAttributeNames\(\)`](#)

---

## **getAttributeNames**

```
java.util.Enumeration getAttributeNames()
```

Returns an `Enumeration` containing the attribute names available within this servlet context. Use the

[`getAttribute\(java.lang.String\)`](#) method with an attribute name to get the value of an attribute.

**Returns:**

an Enumeration of attribute names

**See Also:**

[`getAttribute\(java.lang.String\)`](#)

---

## setAttribute

```
void setAttribute(java.lang.String name,  
                  java.lang.Object object)
```

Binds an object to a given attribute name in this servlet context. If the name specified is already used for an attribute, this method will replace the attribute with the new to the new attribute.

If listeners are configured on the `ServletContext` the container notifies them accordingly.

If a null value is passed, the effect is the same as calling `removeAttribute()`.

Attribute names should follow the same convention as package names. The Java Servlet API specification reserves names matching `java.*`, `javax.*`, and `sun.*`.

**Parameters:**

`name` - a `String` specifying the name of the attribute

`object` - an `Object` representing the attribute to be bound

---

## removeAttribute

```
void removeAttribute(java.lang.String name)
```

Removes the attribute with the given name from the servlet context. After removal, subsequent calls to [`getAttribute\(java.lang.String\)`](#) to retrieve the attribute's value will return `null`.

If listeners are configured on the `ServletContext` the container notifies them accordingly.

**Parameters:**

`name` - a `String` specifying the name of the attribute to be removed

---

## getServletContextName

```
java.lang.String getServletContextName()
```

Returns the name of this web application corresponding to this `ServletContext` as specified in the deployment descriptor for this web application by the `display-name` element.

**Returns:**

The name of the web application or null if no name has been declared in the deployment descriptor.

**Since:**

Servlet 2.3

---

**[Overview](#) [Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)**[PREV CLASS](#) [NEXT CLASS](#)[FRAMES](#) [NO FRAMES](#) [All Classes](#)SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

Copyright © 1999-2002 The Apache Software Foundation. All Rights Reserved.