



XmlStreamWriter Class

The `XmlStreamWriter` class provides methods for writing XML data.

Namespace

[System](#)

Usage

You can use the `XmlStreamWriter` class to programmatically construct an XML document, then use HTTP classes to send the document to an external server.

The `XmlStreamWriter` class is similar to the `XMLStreamWriter` utility class from StAX (Streaming API for XML). StAX is an API to read and write XML documents, originating from the Java programming language community.

Note

The `XmlStreamWriter` class in Apex is based on its counterpart in Java. See [Java XMLStreamWriter class](#).

- [XmlStreamWriter Constructors](#)
- [XmlStreamWriter Methods](#)

See Also

- [Http Class](#)
- [HttpRequest Class](#)
- [HttpResponse Class](#)

XmlStreamWriter Constructors

The following are constructors for `XmlStreamWriter`.

- [XmlStreamWriter\(\)](#)
Creates a new instance of the `XmlStreamWriter` class.

XmlStreamWriter()

Creates a new instance of the `XmlStreamWriter` class.

Signature

```
public XmlStreamWriter()
```

XmlStreamWriter Methods



Returns the XML written by the XmlStreamWriter instance.

- **setDefaultNamespace(uri)**
Binds the specified URI to the default namespace. This URI is bound in the scope of the current START_ELEMENT – END_ELEMENT pair.
- **writeAttribute(prefix, namespaceUri, localName, value)**
Writes an attribute to the output stream.
- **writeCData(data)**
Writes the specified CData to the output stream.
- **writeCharacters(text)**
Writes the specified text to the output stream.
- **writeComment(comment)**
Writes the specified comment to the output stream.
- **writeDefaultNamespace(namespaceUri)**
Writes the specified namespace to the output stream.
- **writeEmptyElement(prefix, localName, namespaceUri)**
Writes an empty element tag to the output stream.
- **writeEndDocument()**
Closes any start tags and writes corresponding end tags to the output stream.
- **writeEndElement()**
Writes an end tag to the output stream, relying on the internal state of the writer to determine the prefix and local name.
- **writeNamespace(prefix, namespaceUri)**
Writes the specified namespace to the output stream.
- **writeProcessingInstruction(target, data)**
Writes the specified processing instruction.
- **writeStartDocument(encoding, version)**
Writes the XML Declaration using the specified XML encoding and version.
- **writeStartElement(prefix, localName, namespaceUri)**
Writes the start tag specified by *localName* to the output stream.

close()

Closes this instance of an XmlStreamWriter and free any resources associated with it.

Signature

```
public Void close()
```

Return Value

Type: Void

getXmlString()

Returns the XML written by the XmlStreamWriter instance.

Signature

```
public String getXmlString()
```

Return Value

Type: String

setDefaultNamespace(uri)



```
public Void setDefaultNamespace(String uri)
```

Parameters

uri

Type: [String](#)

Return Value

Type: Void

writeAttribute(prefix, namespaceUri, localName, value)

Writes an attribute to the output stream.

Signature

```
public Void writeAttribute(String prefix, String namespaceUri, String localName, String value)
```

Parameters

prefix

Type: [String](#)

namespaceUri

Type: [String](#)

localName

Type: [String](#)

Specifies the name of the attribute.

value

Type: [String](#)

Return Value

Type: Void

writeCData(data)

Writes the specified CData to the output stream.

Signature

```
public Void writeCData(String data)
```

Parameters

data

Type: [String](#)

Return Value

Type: Void

writeCharacters(text)

Writes the specified text to the output stream.

Signature



Type: [String](#)

Return Value

Type: Void

writeComment(comment)

Writes the specified comment to the output stream.

Signature

```
public Void writeComment(String comment)
```

Parameters

comment

Type: [String](#)

Return Value

Type: Void

writeDefaultNamespace(namespaceUri)

Writes the specified namespace to the output stream.

Signature

```
public Void writeDefaultNamespace(String namespaceUri)
```

Parameters

namespaceUri

Type: [String](#)

Return Value

Type: Void

writeEmptyElement(prefix, localName, namespaceUri)

Writes an empty element tag to the output stream.

Signature

```
public Void writeEmptyElement(String prefix, String localName, String namespaceUri)
```

Parameters

prefix

Type: [String](#)

localName

Type: [String](#)

Specifies the name of the tag to be written.

namespaceUri

Type: [String](#)

Return Value



Closes any start tags and writes corresponding end tags to the output stream.

Signature

```
public Void writeEndDocument()
```

Return Value

Type: Void

writeEndElement()

Writes an end tag to the output stream, relying on the internal state of the writer to determine the prefix and local name.

Signature

```
public Void writeEndElement()
```

Return Value

Type: Void

writeNamespace(prefix, namespaceUri)

Writes the specified namespace to the output stream.

Signature

```
public Void writeNamespace(String prefix, String namespaceUri)
```

Parameters

prefix

Type: [String](#)

namespaceUri

Type: [String](#)

Return Value

Type: Void

writeProcessingInstruction(target, data)

Writes the specified processing instruction.

Signature

```
public Void writeProcessingInstruction(String target, String data)
```

Parameters

target

Type: [String](#)

data

Type: [String](#)

Return Value

Type: Void



```
public Void writeStartDocument(String encoding, String version)
```

Parameters

encoding

Type: [String](#)

version

Type: [String](#)

Return Value

Type: Void

writeStartElement(prefix, localName, namespaceUri)

Writes the start tag specified by *localName* to the output stream.

Signature

```
public Void writeStartElement(String prefix, String localName, String namespaceUri)
```

Parameters

prefix

Type: [String](#)

localName

Type: [String](#)

namespaceUri

Type: [String](#)

Return Value

Type: Void

DID THIS ARTICLE SOLVE YOUR ISSUE?

Let us know so we can improve!

[Share your feedback](#)



Commerce Cloud
Lightning Design System
Einstein
Quip

Trailhead
Sample Apps
Podcasts
AppExchange

Blog
Salesforce Admins
Salesforce Architects

© Copyright 2025 Salesforce, Inc. [All rights reserved.](#) Various trademarks held by their respective owners. Salesforce, Inc.
Salesforce Tower, 415 Mission Street, 3rd Floor, San Francisco, CA 94105, United States

[Privacy Information](#) [Terms of Service](#) [Legal](#) [Use of Cookies](#) [Trust](#) [Cookie Preferences](#)

[Your Privacy Choices](#) [Responsible Disclosure](#) [Contact](#)