☰　☁　　　　　　　　　　　　　　　　　　　　　🔍　👤

Developers　　　　　　　　　　　　　　　　　　　　　　　　　⌄

URL Class　☰

Apex Reference Guide / System Namespace / OrgLimits Class

URL Class

Represents a uniform resource locator (URL) and provides access to parts of the URL. Enables access to the base URL used to access your Salesforce org.

Namespace

System

Usage

Use the methods of the `System.URL` class to create links to objects in your organization. Such objects can be files, images, logos, or records that you want to include in external emails, in activities, or in Chatter posts. For example, you can create a link to a file uploaded as an attachment to a Chatter post by concatenating the Salesforce base URL with the file ID:

```
// Get a file uploaded through Chatter.
ContentDocument doc = [SELECT Id FROM ContentDocument
        WHERE Title = 'myfile'];
// Create a link to the file.
String fullFileURL = URL.getOrgDomainURL().toExternalForm() +
    '/' + doc.id;
system.debug(fullFileURL);
```

The following example creates a link to a Salesforce record. The full URL is created by concatenating the Salesforce base URL with the record ID.

```
Account acct = [SELECT Id FROM Account WHERE Name = 'Acme' LIMIT 1];
String fullRecordURL = URL.getOrgDomainURL().toExternalForm() + '/' + acct.Id;
```

Example

In this example, the base URL and the full request URL of the current Salesforce server instance are retrieved. Next, a URL pointing to a specific account object is created. Finally, components of the base and full URL are obtained. This example prints out all the results to the debug log output.

```
// Create a new account called Acme that we will create a link for later.
Account myAccount = new Account(Name='Acme');
insert myAccount;

// Get the base URL.
String sfdcBaseURL = URL.getOrgDomainURL().toExternalForm();
System.debug('Base URL: ' + sfdcBaseURL );

// Get the URL for the current request.
String currentRequestURL = URL.getCurrentRequestUrl().toExternalForm();
System.debug('Current request URL: ' + currentRequestURL);

// Create the account URL from the base URL.
String accountURL = URL.getOrgDomainURL().toExternalForm() +
```

```
                              // Get the query string of the current request.
                              System.debug('Query: ' + URL.getCurrentRequestUrl().getQuery());
```

Versioned Behavior Changes

In API version 41.0 and later, Apex URL objects are represented by the `java.net.URI` type, not the `java.net.URL` type. The API version in which the URL object was instantiated determines the behavior of subsequent method calls to the specific instance. Salesforce strongly encourages you to use API 41.0 and later versions for fully RFC-compliant URL parsing that includes proper handling of edge cases of complex URL structures. API 41.0 and later versions also enforce that inputs are valid, RFC-compliant URL or URI strings.

- **URL Constructors**
- **URL Methods**

See Also

- DomainCreator Class

URL Constructors

The following are constructors for `URL`.

- **Url(spec)**
  Creates a new instance of the `URL` class using the specified string representation of the URL.
- **Url(context, spec)**
  Creates a new instance of the `URL` class by parsing the specified spec within the specified context.
- **Url(protocol, host, file)**
  Creates a new instance of the `URL` class using the specified protocol, host, and file on the host. The default port for the specified protocol is used.
- **Url(protocol, host, port, file)**
  Creates a new instance of the `URL` class using the specified protocol, host, port, and file on the host.

Url(spec)

Creates a new instance of the `URL` class using the specified string representation of the URL.

Signature

```
public Url(String spec)
```

Parameters

spec
  Type: String

  The string to parse as a URL.

Url(context, spec)

Creates a new instance of the `URL` class by parsing the specified spec within the specified context.

context

Type: URL

The context in which to parse the specification.

spec

Type: String

The string to parse as a URL.

Usage

The new URL is created from the given context URL and the spec argument as described in RFC2396 "Uniform Resource Identifiers : Generic * Syntax" :

```
<scheme>://<authority><path>?<query>#<fragment>
```

For more information about the arguments of this constructor, see the corresponding URL(java.net.URL, java.lang.String) constructor for Java.

Url(protocol, host, file)

Creates a new instance of the URL class using the specified protocol, host, and file on the host. The default port for the specified protocol is used.

Signature

```
public Url(String protocol, String host, String file)
```

Parameters

protocol

Type: String

The protocol name for this URL.

host

Type: String

The host name for this URL.

file

Type: String

The file name for this URL.

Url(protocol, host, port, file)

Creates a new instance of the URL class using the specified protocol, host, port, and file on the host.

Signature

```
public Url(String protocol, String host, Integer port, String file)
```

Parameters

protocol

Type: String

The protocol name for this URL.

port

Type: Integer

The port number for this URL.

file

Type: String

The file name for this URL.

URL Methods

The following are methods for `URL`.

- **getAuthority()**
  Returns the authority portion of the current URL.

- **getCurrentRequestUrl()**
  Returns the URL of an entire request on a Salesforce instance.

- **getDefaultPort()**
  Returns the default port number of the protocol associated with the current URL.

- **getFile()**
  Returns the file name of the current URL.

- **getFileFieldURL(entityId, fieldName)**
  Returns the download URL for a file attachment.

- **getHost()**
  Returns the host name of the current URL.

- **getOrgDomainUrl()**
  Returns the canonical URL for your org. For example,
  `https://MyDomainName.my.salesforce.com`.

- **getPath()**
  Returns the path portion of the current URL.

- **getPort()**
  Returns the port of the current URL.

- **getProtocol()**
  Returns the protocol name of the current URL, such as, `https`.

- **getQuery()**
  Returns the query portion of the current URL.

- **getRef()**
  Returns the anchor of the current URL.

- **getSalesforceBaseUrl()**
  In API version 59.0 and later, this method is deprecated and versioned out. Use getOrgDomainUrl() to get the canonical URL for your org or use getCurrentRequestUrl() to get the URL of an entire request on a Salesforce instance. Returns the URL of the current connection to the Salesforce org.

- **getUserInfo()**
  Gets the UserInfo portion of the current URL.

- **sameFile(URLToCompare)**
  Compares the current URL with the specified URL object, excluding the fragment component.

- **toExternalForm()**
  Returns a string representation of the current URL.

getAuthority()

Return Value

Type: String

getCurrentRequestUrl()

Returns the URL of an entire request on a Salesforce instance.

Signature

```
public static System.URL getCurrentRequestUrl()
```

Return Value

Type: `System.URL`

Usage

An example of a URL for an entire request is
`https://yourInstance.salesforce.com/apex/myVfPage.apexp`.

getDefaultPort()

Returns the default port number of the protocol associated with the current URL.

Signature

```
public Integer getDefaultPort()
```

Return Value

Type: Integer

Usage

Returns -1 if the URL scheme or the stream protocol handler for the URL doesn't define a default port number.

getFile()

Returns the file name of the current URL.

Signature

```
public String getFile()
```

Return Value

Type: String

getFileFieldURL(entityId, fieldName)

Returns the download URL for a file attachment.

Signature

```
public static String getFileFieldURL(String entityId, String fieldName)
```

Parameters

entityId

Type: String

Specifies the API name of a file field component, such as `AttachmentBody`.

Return Value

Type: String

Usage

Example:

Example

```
String fileURL =
  URL.getFileFieldURL(
     '087000000000123' ,
     'AttachmentBody');
```

getHost()

Returns the host name of the current URL.

Signature

```
public String getHost()
```

Return Value

Type: String

getOrgDomainUrl()

Returns the canonical URL for your org. For example, `https://MyDomainName.my.salesforce.com`.

Signature

```
public static System.Url getOrgDomainUrl()
```

Return Value

Type: `System.URL`

`getOrgDomainUrl()` always returns the login URL for your org, regardless of context. Use that URL when making API calls to your org.

Usage

Use `getOrgDomainUrl()` to interact with Salesforce REST and SOAP APIs in Apex code. Get endpoints for User Interface API calls, for creating and customizing picklist value sets and custom fields, and more.

`getOrgDomainUrl()` can access the domain URL only for the org in which the Apex code is running.

You don't need a RemoteSiteSetting for your org to interact with the Salesforce APIs using domain URLs retrieved with this method.

Example

This example uses the Salesforce REST API to get organization limit values. For information on limits, see Limits in the *REST API Developer Guide*.

```
    req.setMethod('GET');
    req.setHeader('Authorization', 'Bearer ' + UserInfo.getSessionId());
    HttpResponse res = h.send(req);
```

See Also

- *Lightning Aura Components Developer Guide*: Making API Calls from Apex
- *User Interface API Developer Guide*: Get Default Values to Clone a Record
- *User Interface API Developer Guide*: Get Values for a Picklist Field
- *User Interface API Developer Guide*: User Inteface API Resources

getPath()

Returns the path portion of the current URL.

Signature

```
public String getPath()
```

Return Value

Type: String

getPort()

Returns the port of the current URL.

Signature

```
public Integer getPort()
```

Return Value

Type: Integer

getProtocol()

Returns the protocol name of the current URL, such as, `https`.

Signature

```
public String getProtocol()
```

Return Value

Type: String

getQuery()

Returns the query portion of the current URL.

Signature

```
public String getQuery()
```

Return Value

Type: String

Usage

Returns the anchor of the current URL.

Signature

```
public String getRef()
```

Return Value

Type: String

Usage

Returns `null` if no query portion exists.

getSalesforceBaseUrl()

In API version 59.0 and later, this method is deprecated and versioned out. Use getOrgDomainUrl() to get the canonical URL for your org or use getCurrentRequestUrl() to get the URL of an entire request on a Salesforce instance. Returns the URL of the current connection to the Salesforce org.

Signature

```
public static System.URL getSalesforceBaseUrl()
```

Return Value

Type: `System.URL`

Returns the URL for the current connection: for example, `https://MyDomainName.my.salesforce.com` or `https://MyDomainName.lightning.force.com`.

See Also

- getOrgDomainUrl()

getUserInfo()

Gets the UserInfo portion of the current URL.

Signature

```
public String getUserInfo()
```

Return Value

Type: String

Usage

Returns `null` if no UserInfo portion exists.

sameFile(URLToCompare)

Compares the current URL with the specified URL object, excluding the fragment component.

Signature

```
public Boolean sameFile(System.URL URLToCompare)
```

Parameters

URLToCompare

Returns `true` if both URL objects reference the same remote resource; otherwise, returns `false`.

Usage

For more information about the syntax of URIs and fragment components, see [RFC3986](#).

toExternalForm()

Returns a string representation of the current URL.

Signature

```
public String toExternalForm()
```

Return Value

Type: String

DID THIS ARTICLE SOLVE YOUR ISSUE?
Let us know so we can improve!

[Share your feedback]

DEVELOPER CENTERS

Heroku

MuleSoft

Tableau

Commerce Cloud

Lightning Design System

Einstein

Quip

POPULAR RESOURCES

Documentation

Component Library

APIs

Trailhead

Sample Apps

Podcasts

AppExchange

COMMUNITY

Trailblazer Community

Events and Calendar

Partner Community

Blog

Salesforce Admins

Salesforce Architects

Privacy Information   Terms of Service   Legal   Use of Cookies   Trust   Cookie Preferences

Your Privacy Choices   Responsible Disclosure   Contact