



System Namespace

The `System` namespace provides classes and methods for core Apex functionality.

The following are the classes in the `System` namespace.

- **AccessLevel Class**
Defines the different modes, such as system or user mode, that Apex database operations execute in.
- **AccessType Enum**
Specifies the access check type for the fields of an `sObject`.
- **Address Class**
Contains methods for accessing the component fields of address compound fields.
- **Answers Class**
Represents zone answers.
- **ApexPages Class**
Use `ApexPages` to add and check for messages associated with the current page, as well as to reference the current page.
- **Approval Class**
Contains methods for processing approval requests and setting approval-process locks and unlocks on records.
- **Assert Class**
Contains methods to assert various conditions with test methods, such as whether two values are the same, a condition is true, or a variable is null.
- **AsyncInfo Class**
Provides methods to get the current stack depth, maximum stack depth, and the minimum queueable delay for Queueable transactions, and to determine if maximum stack depth is set.
- **AsyncOptions Class**
Contains maximum stack depths for queueable transactions and the minimum queueable delay in minutes. Passed as parameter to the `System.enqueueJob()` method to define a unique queueable job signature, the maximum stack depth for queueable transactions and the minimum queueable delay in minutes.
- **Blob Class**
Contains methods for the Blob primitive data type.
- **Boolean Class**
Contains methods for the Boolean primitive data type.
- **BusinessHours Class**
Use the `BusinessHours` methods to set the business hours at which your customer support team operates.
- **CallbackStatus Enum**
Specifies the status of asynchronous requests to an external system.
- **Callable Interface**
Enables developers to use a common interface to build loosely coupled integrations between Apex classes or triggers, even for code in separate packages. Agreeing upon a common interface enables developers from different companies or different departments to build upon one another's solutions. Implement this interface to enable the broader community, which might have different solutions than the ones you had in mind, to extend your code's functionality.
- **Cases Class**
Use the `Cases` class to interact with case records.



- **Comparable Interface**
Adds sorting support for Lists that contain non-primitive types, that is, Lists of user-defined types. Your implementation must explicitly handle null inputs in the `compareTo()` method to avoid a null pointer exception.
- **Comparator Interface**
Implement different sort orders with the Comparator interface's `compare()` method, and pass the Comparator as a parameter to `List.sort()`. Your implementation must explicitly handle null inputs in the `compare()` method to avoid a null pointer exception.
- **Continuation Class**
Use the `Continuation` class to make callouts asynchronously to a SOAP or REST Web service.
- **Cookie Class**
The `Cookie` class lets you access cookies for your Salesforce site using Apex.
- **Crypto Class**
Provides methods for creating digests, message authentication codes, and signatures, as well as encrypting and decrypting information.
- **Custom Metadata Type Methods**
Custom metadata types are customizable, deployable, packageable, and upgradeable application metadata. All custom metadata is exposed in the application cache, which allows access without repeated queries to the database. The metadata is then available for formula fields, validation rules, flows, Apex, and SOAP API. All methods are static.
- **Custom Settings Methods**
Custom settings are similar to custom objects and enable application developers to create custom sets of data, as well as create and associate custom data for an organization, profile, or specific user. All custom settings data is exposed in the application cache, which enables efficient access without the cost of repeated queries to the database. This data is then available for formula fields, validation rules, flows, Apex, and the SOAP API.
- **Database Class**
Contains methods for creating and manipulating data.
- **Date Class**
Contains methods for the Date primitive data type.
- **Datetime Class**
Contains methods for the Datetime primitive data type.
- **Decimal Class**
Contains methods for the Decimal primitive data type.
- **Domain Class**
Represents an existing domain hosted by Salesforce that serves the org or its content. Contains methods to obtain information about these domains, such as the domain type, My Domain name, and sandbox name.
- **DomainCreator Class**
Use the `DomainCreator` class to return a hostname specific to the org. For example, get the org's Visualforce hostname. Values are returned as a hostname, such as `MyDomainName.lightning.force.com`.
- **DomainParser Class**
Use the `DomainParser` class to parse a domain that Salesforce hosts for the org and extract information about the domain.
- **DomainType Enum**
Specifies the domain type for a `System.Domain`.
- **Double Class**
Contains methods for the Double primitive data type.
- **EmailMessages Class**
Use the methods in the `EmailMessages` class to interact with emails and email threading.
- **EncodingUtil Class**
Use the methods in the `EncodingUtil` class to encode and decode URL strings, and



you can define your own enum.

- **EventBus Class**
Contains methods for publishing platform events.
- **Exception Class and Built-In Exceptions**
An exception denotes an error that disrupts the normal flow of code execution. You can use Apex built-in exceptions or create custom exceptions. All exceptions have common methods.
- **ExternalServiceTest Class**
Provides methods to test an external service's asynchronous callouts, enables sending a mock request, asserts the expected request payload, then triggers the mocked external service's asynchronous callback response.
- **FlexQueue Class**
Contains methods that reorder batch jobs in the Apex flex queue.
- **FeatureManagement Class**
Use the methods in the `System.FeatureManagement` class to check and modify the values of feature parameters, and to show or hide custom objects and custom permissions in your subscribers' orgs.
- **Formula Class**
Contains methods to get a builder for creating a formula instance and to update all formula fields on the input SObjects.
- **FormulaRecalcFieldError Class**
The return type of the `FormulaRecalcResult.getErrors` method.
- **FormulaRecalcResult Class**
The return type of the `Formula.recalculateFormulas` method.
- **Http Class**
Use the `Http` class to initiate an HTTP request and response.
- **HttpCalloutMock Interface**
Enables sending fake responses when testing HTTP callouts.
- **HttpRequest Class**
Use the `HttpRequest` class to programmatically create HTTP requests like GET, POST, PATCH, PUT, and DELETE.
- **HttpResponse Class**
Use the `HttpResponse` class to handle the HTTP response returned by the `Http` class.
- **Id Class**
Contains methods for the ID primitive data type.
- **Ideas Class**
Represents zone ideas.
- **InstallHandler Interface**
Enables custom code to run after a managed package installation or upgrade.
- **Integer Class**
Contains methods for the Integer primitive data type.
- **JSON Class**
Contains methods for serializing Apex objects into JSON format and deserializing JSON content that was serialized using the `serialize` method in this class.
- **JSONGenerator Class**
Contains methods used to serialize objects into JSON content using the standard JSON encoding.
- **JSONParser Class**
Represents a parser for JSON-encoded content.
- **JSONToken Enum**
Contains all token values used for parsing JSON content.
- **Label Class**
Provides methods to retrieve a custom label or to check if translation exists for a label in a specific language and namespace. Label names are dynamically resolved at run time,



- **List Class**
Contains methods for the List collection type.
- **Location Class**
Contains methods for accessing the component fields of geolocation compound fields.
- **LoggingLevel Enum**
Specifies the logging level for the `System.debug` method.
- **Long Class**
Contains methods for the Long primitive data type.
- **Map Class**
Contains methods for the Map collection type.
- **Matcher Class**
Matchers use Patterns to perform match operations on a character string.
- **Math Class**
Contains methods for mathematical operations.
- **Messaging Class**
Contains messaging methods used when sending a single or mass email.
- **MultiStaticResourceCalloutMock Class**
Utility class used to specify a fake response using multiple resources for testing HTTP callouts.
- **Network Class**
Manage Experience Cloud sites.
- **Object Class**
Contains methods that are implemented by all Apex types.
- **OrgLimit Class**
Contains methods that provide the name, maximum value, and current value of an org limit.
- **OrgLimits Class**
Contains methods that provide a list or map of all OrgLimit instances for Salesforce your org, such as SOAP API requests, Bulk API requests, and Streaming API limits.
- **PageReference Class**
A PageReference is a reference to an instantiation of a page. Among other attributes, PageReferences consist of a URL and a set of query parameter names and values.
- **Packaging Class**
Contains a method for obtaining information about managed and unlocked packages.
- **Pattern Class**
Represents a compiled representation of a regular expression.
- **Queueable Interface**
Enables the asynchronous execution of Apex jobs that can be monitored.
- **QueueableContext Interface**
Represents the parameter type of the `execute()` method in a class that implements the `Queueable` interface and contains the job ID. This interface is implemented internally by Apex.
- **QueueableDuplicateSignature Class**
Used in the `AsyncOptions` class to store the queueable job signature in the `DuplicateSignature` property.
- **QueueableDuplicateSignature.Builder Class**
Build a unique signature for your queueable job using this inner builder class. The `build()` class method builds a `QueueableDuplicateSignature` object, with input from the `addId()`, `addInteger()`, and `addString()` methods. Use the `DuplicateSignature` property in the `AsyncOptions` class to store the queueable job signature. Enqueue your job by using the `System.enqueueJob()` with the `AsyncOptions` parameter.
- **QuickAction Class**
Use Apex to request and process actions on objects that allow custom fields, on objects that appear in a Chatter feed, or on objects that are available globally.



operations in your remote Objects override methods.

- **Request Class**
Contains methods to obtain the request ID and Quiddity value of the current Salesforce request.
- **ResetPasswordResult Class**
Represents the result of a password reset.
- **RestContext Class**
Contains the `RestRequest` and `RestResponse` objects.
- **RestRequest Class**
Use the `System.RestRequest` class to access and pass request data in a RESTful Apex method.
- **RestResponse Class**
Represents an object used to pass data from an Apex RESTful Web service method to an HTTP response.
- **SandboxPostCopy Interface**
To make your sandbox environment business ready, automate data manipulation or business logic tasks. Extend this interface and add methods to perform post-copy tasks, then specify the class during sandbox creation.
- **Schedulable Interface**
The class that implements this interface can be scheduled to run at different intervals.
- **SchedulableContext Interface**
Represents the parameter type of a method in a class that implements the `Schedulable` interface and contains the scheduled job ID. This interface is implemented internally by Apex.
- **Schema Class**
Contains methods for obtaining schema describe information.
- **Search Class**
Use the methods of the `Search` class to perform dynamic SOSL queries.
- **Security Class**
Contains methods to securely implement Apex applications.
- **SelectOption Class**
A `SelectOption` object specifies one of the possible values for a Visualforce `selectCheckboxes`, `selectList`, or `selectRadio` component.
- **Set Class**
Represents a collection of unique elements with no duplicate values.
- **Site Class**
Use the `site` Class to manage your sites. Change, reset, validate, and check the expiration of passwords. Create site users, person accounts, and portal users. Get the admin email and ID. Get various URLs, the path prefix, the ID, the template, and the type of the site. Log in to the site.
- **SObject Class**
Contains methods for the `sObject` data type.
- **SObjectAccessDecision Class**
Contains the results of a call to the `Security.stripInaccessible` method and methods to retrieve those results.
- **SoqlStubProvider Class**
Contains a method to create a mock test class for handling SOQL query responses for Data Cloud data model objects (DMOs).
- **StaticResourceCalloutMock Class**
Utility class used to specify a fake response for testing HTTP callouts.
- **String Class**
Contains methods for the `String` primitive data type.
- **StubProvider Interface**
`StubProvider` is a callback interface that you can use as part of the Apex stub API to



Jobs.

- **Test Class**
Contains methods related to Apex tests.
- **Time Class**
Contains methods for the Time primitive data type.
- **TimeZone Class**
Represents a time zone. Contains methods for creating a new time zone and obtaining time zone properties, such as the time zone ID, offset, and display name.
- **Trigger Class**
Use the `Trigger` class to access run-time context information in a trigger, such as the type of trigger or the list of `sObject` records that the trigger operates on.
- **TriggerOperation Enum**
`System.TriggerOperation` enum values are associated with trigger events.
- **Type Class**
Contains methods for getting the Apex type that corresponds to an Apex class and for instantiating new types.
- **UninstallHandler Interface**
Enables custom code to run after a managed package is uninstalled.
- **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.
- **UserInfo Class**
Contains methods for obtaining information about the context user.
- **UserManagement Class**
Contains methods to manage end users, for example, to register their verification methods, verify their identity, or remove their personal information.
- **UUID Class**
Contains methods to randomly generate a version 4 universally unique identifier (UUID), compare UUIDs, and convert UUID instance to a string.
- **Version Class**
Use the `Version` methods to get the version of a first-generation managed package, and to compare package versions.
- **WebServiceCallout Class**
Enables making callouts to SOAP operations on an external Web service. This class is used in the Apex stub class that is auto-generated from a WSDL.
- **WebServiceMock Interface**
Enables sending fake responses when testing Web service callouts of a class auto-generated from a WSDL.
- **XmlStreamReader Class**
The `XmlStreamReader` class provides methods for forward, read-only access to XML data. You can pull data from XML or skip unwanted events. You can parse nested XML content that's up to 50 nodes deep.
- **XmlStreamWriter Class**
The `XmlStreamWriter` class provides methods for writing XML data.

DID THIS ARTICLE SOLVE YOUR ISSUE?

Let us know so we can improve!

[Share your feedback](#)



MuleSoft
Tableau
Commerce Cloud
Lightning Design System
Einstein
Quip

Component Library
APIs
Trailhead
Sample Apps
Podcasts
AppExchange

Events and Calendar
Partner Community
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](#)