



# 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.

## Namespace

[System](#)

## Usage

Use the methods in the `System.JSON` class to perform round-trip JSON serialization and deserialization of Apex objects.

### See Also

- [Apex Developer Guide: Roundtrip Serialization and Deserialization](#)

## JSON Methods

The following are methods for `JSON`. All methods are static.

- [createGenerator\(prettyPrint\)](#)  
Returns a new JSON generator.
- [createParser\(jsonString\)](#)  
Returns a new JSON parser.
- [deserialize\(jsonString, apexType\)](#)  
Deserializes the specified JSON string into an Apex object of the specified type.
- [deserializeStrict\(jsonString, apexType\)](#)  
Deserializes the specified JSON string into an Apex object of the specified type.
- [deserializeUntyped\(jsonString\)](#)  
Deserializes the specified JSON string into collections of primitive data types.
- [serialize\(objectToSerialize\)](#)  
Serializes Apex objects into JSON content.
- [serialize\(objectToSerialize, suppressApexObjectNulls\)](#)  
Suppresses `null` values when serializing Apex objects into JSON content.
- [serializePretty\(objectToSerialize\)](#)  
Serializes Apex objects into JSON content and generates indented content using the pretty-print format.
- [serializePretty\(objectToSerialize, suppressApexObjectNulls\)](#)  
Suppresses `null` values when serializing Apex objects into JSON content and generates indented content using the pretty-print format.

### createGenerator(prettyPrint)

Returns a new JSON generator.

#### Signature



Type: [Boolean](#)

Determines whether the JSON generator creates JSON content in pretty-print format with the content indented. Set to `true` to create indented content.

#### Return Value

Type: [System.JSONGenerator](#)

## createParser(jsonString)

Returns a new JSON parser.

#### Signature

```
public static System.JSONParser createParser(String jsonString)
```

#### Parameters

##### *jsonString*

Type: [String](#)

The JSON content to parse.

#### Return Value

Type: [System.JSONParser](#)

## deserialize(jsonString, apexType)

Deserializes the specified JSON string into an Apex object of the specified type.

#### Signature

```
public static Object deserialize(String jsonString, System.Type apexType)
```

#### Parameters

##### *jsonString*

Type: [String](#)

The JSON content to deserialize.

##### *apexType*

Type: [System.Type](#)

The Apex type of the object that this method creates after deserializing the JSON content.

#### Return Value

Type: `Object`

#### Usage

If the JSON content contains attributes not present in the `System.Type` argument, such as a missing field or object, deserialization fails in some circumstances. When deserializing JSON content into a custom object or an `sObject` using Salesforce API version 34.0 or earlier, this method throws a runtime exception when passed extraneous attributes. When deserializing JSON content into an Apex class in any API version, or into an object in API version 35.0 or later, no exception is thrown. When no exception is thrown, this method ignores extraneous attributes and parses the rest of the JSON content.

#### Example



```
'100.1', Decimal.class);  
System.assertEquals(n, 100.1);
```

## deserializeStrict(jsonString, apexType)

Deserializes the specified JSON string into an Apex object of the specified type.

### Signature

```
public static Object deserializeStrict(String jsonString, System.Type apexType)
```

### Parameters

#### *jsonString*

Type: [String](#)

The JSON content to deserialize.

#### *apexType*

Type: [System.Type](#)

The Apex type of the object that this method creates after deserializing the JSON content.

### Return Value

Type: Object

### Usage

All attributes in the JSON string must be present in the specified type. If the JSON content contains attributes not present in the `System.Type` argument, such as a missing field or object, deserialization fails in some circumstances. When deserializing JSON content with extraneous attributes into an Apex class, this method throws an exception in all API versions. However, no exception is thrown when you use this method to deserialize JSON content into a custom object or an sObject.

### Example

The following example deserializes a JSON string into an object of a user-defined type represented by the `Car` class, which this example also defines.

```
public class Car {  
    public String make;  
    public String year;  
}  
  
public void parse() {  
    Car c = (Car)JSON.deserializeStrict(  
        '{"make":"SFDC","year":"2020"}',  
        Car.class);  
    System.assertEquals(c.make, 'SFDC');  
    System.assertEquals(c.year, '2020');  
}
```

## deserializeUntyped(jsonString)

Deserializes the specified JSON string into collections of primitive data types.

### Signature

```
public static Object deserializeUntyped(String jsonString)
```



The JSON content to deserialize.

### Return Value

Type: Object

### Example

The following example deserializes a JSON representation of an appliance object into a map that contains primitive data types and further collections of primitive types. It then verifies the deserialized values.

```
String jsonInput = '{\n' +
    '  "description" : "An appliance",\n' +
    '  "accessories" : [ "powerCord", ' +
    '    { "right": "door handle1", ' +
    '      "left": "door handle2" } ],\n' +
    '  "dimensions" : ' +
    '    { "height" : 5.5 , ' +
    '      "width" : 3.0 , ' +
    '      "depth" : 2.2 },\n' +
    '  "type" : null,\n' +
    '  "inventory" : 2000,\n' +
    '  "price" : 1023.45,\n' +
    '  "isShipped" : true,\n' +
    '  "modelNumber" : "123"\n' +
    '  }';

Map<String, Object> m =
    (Map<String, Object>)
        JSON.deserializeUntyped(jsonInput);

System.assertEquals(
    'An appliance', m.get('description'));

List<Object> a =
    (List<Object>)m.get('accessories');
System.assertEquals('powerCord', a[0]);
Map<String, Object> a2 =
    (Map<String, Object>)a[1];
System.assertEquals(
    'door handle1', a2.get('right'));
System.assertEquals(
    'door handle2', a2.get('left'));

Map<String, Object> dim =
    (Map<String, Object>)m.get('dimensions');
System.assertEquals(
    5.5, dim.get('height'));
System.assertEquals(
    3.0, dim.get('width'));
System.assertEquals(
    2.2, dim.get('depth'));

System.assertEquals(null, m.get('type'));
System.assertEquals(
    2000, m.get('inventory'));
System.assertEquals(
    1023.45, m.get('price'));
System.assertEquals(
    true, m.get('isShipped'));
System.assertEquals(
    '123', m.get('modelNumber'));
```

## serialize(objectToSerialize)

Serializes Apex objects into JSON content.



### *objectToSerialize*

Type: Object

The Apex object to serialize.

### Return Value

Type: [String](#)

### Example

The following example serializes a new `Datetime` value.

```
Datetime dt = Datetime.newInstance(  
    Date.newInstance(  
        2011, 3, 22),  
    Time.newInstance(  
        1, 15, 18, 0));  
String str = JSON.serialize(dt);  
System.assertEquals(  
    '"2011-03-22T08:15:18.000Z"',  
    str);
```

## **serialize(objectToSerialize, suppressApexObjectNulls)**

Suppresses `null` values when serializing Apex objects into JSON content.

### Signature

```
public static String serialize(Object objectToSerialize, Boolean suppressApexObjectNulls)
```

### Parameters

#### *objectToSerialize*

Type: Object

The Apex object to serialize.

#### *suppressApexObjectNulls*

Type: [Boolean](#)

If true, remove `null` values before serializing the JSON object.

#### Note

This parameter doesn't apply to sObjects retrieved via SOQL.

### Return Value

Type: [String](#)

### Usage

This method allows you to specify whether to suppress `null` values when serializing Apex objects into JSON content.

## **serializePretty(objectToSerialize)**



```
public static String serializePretty(Object objectToSerialize)
```

#### Parameters

##### *objectToSerialize*

Type: `Object`

The Apex object to serialize.

#### Return Value

Type: `String`

## **serializePretty(objectToSerialize, suppressApexObjectNulls)**

Suppresses `null` values when serializing Apex objects into JSON content and generates indented content using the pretty-print format.

#### Signature

```
public static String serializePretty(Object objectToSerialize, Boolean suppressApexObjectNulls)
```

#### Parameters

##### *objectToSerialize*

Type: `Object`

The Apex object to serialize.

##### *suppressApexObjectNulls*

Type: `Boolean`

If true, remove `null` values before serializing the JSON object.



#### Note

This parameter doesn't apply to sObjects retrieved via SOQL.

#### Return Value

Type: `String`

#### DID THIS ARTICLE SOLVE YOUR ISSUE?

Let us know so we can improve!

[Share your feedback](#)



#### DEVELOPER CENTERS

[Heroku](#)

[MuleSoft](#)

#### POPULAR RESOURCES

[Documentation](#)

[Component Library](#)

#### COMMUNITY

[Trailblazer Community](#)

[Events and Calendar](#)



Quip

AppExchange

© 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](#)