

Salesforce Developer's Manual: uuid_Gen Class

The `uuid_Gen` class provides a set of methods for generating and validating Universally Unique Identifiers (UUIDs) in Salesforce. UUIDs are commonly used to uniquely identify records and objects within an application.

This manual will guide Salesforce developers on how to use the `uuid_Gen` class effectively in their code. It covers the available methods, their parameters, and their return types.

Class Overview

The `uuid_Gen` class is defined with the following attributes:

apexCopy code

```
global with sharing class uuid_Gen {  
    // Class methods and variables  
}
```

The class has a global access modifier, allowing it to be accessed from any Apex class or trigger.

Class Methods

Method: isUuid

apexCopy code

```
global static Boolean isUuid(String uuid)  
global static Boolean isUuid(Integer version, String uuid)
```

The `isUuid` method is used to check the validity of a UUID. It accepts a UUID string as a parameter and returns a Boolean value indicating whether the UUID is valid.

- `uuid` (String): The UUID to be validated.
- `version` (Integer): (Optional) The UUID version to be checked. If specified, the method checks if the provided UUID matches the specified version. If not specified, it checks the validity of the UUID without considering the version.

Return Type: Boolean

Method: v1

apexCopy code

```
global static String v1()
global static String v1(Datetime dt)
global static String v1(List<Integer> node)
global static String v1(String node)
global static String v1(Datetime dt, String node)
global static String v1(Datetime dt, List<Integer> node)
```

The `v1` method generates a Version 1 UUID. Version 1 UUIDs are based on the timestamp and MAC address of the generating node.

- `dt` (Datetime): (Optional) The timestamp to be used for generating the UUID. If not specified, the current timestamp is used.
- `node` (String or List of Integer): (Optional) The MAC address or node identifier to be used. It can be either a MAC address represented as a string or a list of six integers representing the node identifier.

Return Type: String

Method: v3

apexCopy code

```
global static String v3(Namespace namespace, String value)
global static String v3(Namespace namespace, Blob value)
global static String v3(String namespace, String value)
global static String v3(String namespace, Blob value)
```

The `v3` method generates a Version 3 UUID using the provided namespace and value. Version 3 UUIDs are generated using MD5 hashing.

- `namespace` (Namespace or String): The namespace to be used for generating the UUID. It can be one of the following enum values: `DNS` , `URL` , `OID` , or `X500` , or a custom string representing the namespace.
- `value` (String or Blob): The value to be used for generating the UUID. It can be a string or a blob.

Return Type: String

Method: v4

apexCopy code

```
global static String v4()  
global static List<String> v4(Integer amount)
```

The `v4` method generates one or more Version 4 UUIDs. Version 4 UUIDs are randomly generated.

- `amount` (Integer): (Optional) The number of UUIDs to generate. If not specified, it generates a single UUID.

Return Type: String or List

Method: v5

apexCopy code

```
global static String v5(Namespace namespace, String value)  
global static String v5(Namespace namespace, Blob value)  
global static String v5(String namespace, String value)  
global static String v5(String namespace, Blob value)
```

The `v5` method generates a Version 5 UUID using the provided namespace and value. Version 5 UUIDs are generated using SHA-1 hashing.

- `namespace` (Namespace or String): The namespace to be used for generating the UUID. It can be one of the following enum values: `DNS` , `URL` , `OID` , or `X500` , or a custom string representing the namespace.
- `value` (String or Blob): The value to be used for generating the UUID. It can be a string or a blob.

Return Type: String

Method: nil

apexCopy code

```
global static String nil()
```

The `nil` method returns a nil UUID, which is a special UUID with all zeros.

Return Type: String

Usage Example

Here's an example of how you can use the `uuid_Gen` class in your code:

apexCopy code

```
// Check if a UUID is valid
Boolean isValid = uuid_Gen.isUuid('6ba7b810-9dad-11d1-80b4-00c04fd430c8');
System.debug('Is valid UUID: ' + isValid);

// Generate a Version 1 UUID
String v1Uuid = uuid_Gen.v1();
System.debug('Version 1 UUID: ' + v1Uuid);

// Generate a Version 3 UUID
String v3Uuid = uuid_Gen.v3(uuid_Gen.Namespace.DNS, 'example.com');
System.debug('Version 3 UUID: ' + v3Uuid);

// Generate a Version 4 UUID
String v4Uuid = uuid_Gen.v4();
System.debug('Version 4 UUID: ' + v4Uuid);

// Generate a Version 5 UUID
String v5Uuid = uuid_Gen.v5(uuid_Gen.Namespace.URL, 'https://www.example.com');
System.debug('Version 5 UUID: ' + v5Uuid);

// Get a nil UUID
String nilUuid = uuid_Gen.nil();
System.debug('Nil UUID: ' + nilUuid);
```

This example demonstrates various methods available in the `uuid_Gen` class, including UUID validation, generation of different versions of UUIDs, and obtaining a nil UUID.

Conclusion

The `uuid_Gen` class provides Salesforce developers with a convenient way to generate and validate UUIDs. By following the instructions outlined in this manual, you can easily incorporate UUID functionality into your Salesforce code.