API Reference

Create Profile Extension Schema

5/12/2025

Create Profile Extension Schema

	POST /metadata/profiles/extensions
	Available since: 8.0.100.00

Contents

Description

Creates a profile extension schema resource, which is mandatory to query and use profile extension schemas. Your application can create profile extension schemas when the UCS is in Production mode, but Genesys recommends to create extensions when the server is in maintenance mode.

- Read more about UCS Modes for Context Services.
- Read more about Extensions.

⚠️ 8.0.1 does not support multi-valued extensions.

Operation

Prerequisites:

- The server is in Maintenance mode.

Create Profile Extensions Schema

ID	CV.WS.PROF.META.4		
Method	POST		
URL	/metadata/profiles/extensions		
Parameter	Type	Mandatory	Description
Body: The schema to create.<ref>This is the Profile Extension Schema to create.</ref>			
name	string	yes	The unique, case-insensitive name of the extension. The name must start with a letter, and can be followed with letters, numbers, or underscores. The name is restricted to a maximum of 26 characters (Maximum RDBMS shared limit on creation of index).
type	token • "single-valued" • "multi-valued"	yes	Extensions come in the following forms: • Single-valued: a single extension record is associated with a given customer.

ID	CV.WS.PROF.META.4		
			• Multi-valued: multiple extension records can be associated with a given customer.
attributes	Attribute Schema[]	no	The array of *zero or more* attributes. Empty when type="classifier".
unique	string[]	no	Lists attributes which are unique in the scope of a given customer. The customer profile cannot include several extension records with identical values for these attributes. For example, let's consider a contact extension with the attributes "phone_num", "ext", "time_of_day". To ensure that a given customer does not have two contact extensions with the same phone number, set the following in the extension schema: unique = ["phone_num", "ext"]

Response

The Context Management Service API answers with HTTP codes for every request. The following table shows the correct response for a successful request. See HTTP Response Codes and Errors for further details on the possible codes that this operation can return.

Response

HTTP code	201
HTTP message	Created
Header	Location: **${base_uri}**/profiles/ extensions/**${extension-name}** where

HTTP code	201
	• ${extension-name} is the name of the extension provided in the request's entity body. • ${base_uri} is the base URI of Context Services.
Body	{ "name": "**${extension-name}**"} where: • ${extension-name} is the name of the created extension.

Example

Operation

```
POST http://ucsserver.mycompany.com:8080/cms/profiles/extensions
{
  "name":"Address",
  "type":"single-valued",
  "attributes": [
      {"name":"AddressType","type":"integer","default":0},
      {"name":"Address","type":"string","length":256},
      {"name":"City","type":"string","length":32},
      {"name":"County","type":"string","length":32},
      {"name":"PostCode","type":"string", "length":10},
      {"name":"Country","type":"string","length":32}
      ]
  }
```

Result

```
201 Created
Content-Location        http://ucsserver.mycompany.com:8080/cms/profiles/extensions/Address
{"name":"Address"}
```