API Provisioning Plugin

The API Provisioning Plugin provisions CO Person records to a RESTful or messaging endpoint.

Operations

Registry CO Person Transaction	Message Action
Add	Send person object
Edit	Send person object
Enter Grace Period	Send person object
Expiration / Becomes Inactive	Send person object
Unexpire / Becomes Active	Send person object
Delete	Send person object with deleted flag*
Manual Provision	Send person object

*See URL Construction and Message Format below for additional information

Registry CO Group Transaction	Message Action
Add	None
Edit	None
Delete	None
Manual Provision	None

Installation

This is a non-core plugin, see Installing and Enabling Registry Plugins for more information.

Configuration

Currently, the ApiProvisioner Plugin only supports sending a JSON message (format defined below) to a designated RESTful endpoint. Future versions may support additional formats and/or other protocols.

1. Define a new Server in Registry.
    a. *Servers > Add a New Server*
    b. Server Type: HTTP
    c. On the next page, configure the Server as follows
        i. Set the *Server URL* to the endpoint where the message will be set. More details below.
        ii. Set the *Username* and *Password* as needed.
2. Configure a new Provisioning Target in Registry.
    a. *Configuration > Provisioning Targets > Add Provisioning Target*
        i. *Plugin*: ApiProvisioner
        ii. On the next page, set the following:
            1. Select the *Server* created in the previous step.
            2. Select the *Protocol Mode*
            3. Set the *Identifier Type* that will be used as the person's unique identifier. This will be included in the JSON message body (described below).

4. *Include Attributes* must be set, not including attributes is not yet supported (CO-1863, CO-2004).

Plugin Operation

Authentication

The *Username* and *Password* are sent to the server using Basic Auth. As such, the use of HTTPS is strongly recommended.

URL Construction

HTTP POST

The URL is constructed as the *Server URL* with an appended trailing slash (`/`).

Example
POST https://myserver.nil/rest/upload**/**

HTTP PUT

The URL is constructed as the *Server URL* with an appended trailing slash (`/`) and the Identifier of the configured type.

On a delete action, the HTTP method will be `DELETE` instead of `PUT`.

Example
PUT https://myserver.nil/rest/people**/A12345**
DELETE https://myserver.nil/rest/people**/A12345**

Message Format

The message is sent to the URL endpoint with a `Content-Type` of `application/json`.

The message body is described using JSON Schema:

- Latest Format (Provisioner API v1.0.0, develop)
- Provisioner API v1.0.0 (Registry v4.0.0)

⚠ When the Protocol Mode is *HTTP PUT* and the action being handled is Delete CO Person, no message body is sent.

Expected Response

The HTTP Response Code is examined for successful processing of the message. Any 200 series response is considered successful receipt of the message.

The ApiProvisioner Plugin does not currently expect any message body in the response from the endpoint.

See Also

- cm_co_api_provisioner_targets