Upgrade Guide | PUBLIC
Document Version: 2204a – 2022-04-19

Migration Guide

THE BEST RUN SAP

Content

1 Overview. 3
1.1 Migration Services: Location. 4

1 Overview

Migrating data from existing scenarios to the new data model of SAP IoT

Introduction

SAP IoT provides a new and more powerful service for managing locations. This new service replaces the old location service that dates back to our now outdated offering SAP IoT Application Enablement. As long as the deprecation phase is not completed, both services (old and new) are available and you can decide which service you want to use to assign locations to your things.

To make life easy for you, SAP offers a migration path for your location-related data. The migration consists of the following phases:

General Process

In principle, the migration of data from the source environment to the target environment follows a multi-step approach:

1. Initial Data Replication: The migration process starts with an initial data replication run triggered by SAP. During this run, all data that exists in the source environment at this point in time is transferred into the target environment.
2. Replication Phase: All data is created, updated, and deleted in the deprecated location service the same way as it used to be in earlier releases. In addition, all these changes to data are automatically replicated into the new location service. During this phase, the source environment is still considered the leading environment. That is, if users change a location in new service and then update it in the old service, the changes done to the new service will be overwritten.

   > **i** Note
   >
   > Data replication is a unidirectional process. That is, replication takes place from source to target, but **not** vice versa.

3. Proxy Phase: As soon as a customer has gained the impression that the data in the target environment properly reflects the situation in the source environment, it is time for switching from replication phase to proxy phase. In the proxy phase, all services and endpoints remain the same as before. However, in the backend, a redirection mechanism is activated that directs all data accesses exclusively to the target landscape.

   > **i** Note
   >
   > The proxy phase serves the purpose of smoothing the transition from the old location service to the new geolocation service. Custom coding using the old location service can be adapted in this phase. If no custom coding exists, we recommend using the new geolocation service and its APIs right from the start. For existing customers, note that for reading location data you need the scopes of the new service.
   >
   > During proxy phase, you should be aware of the following potential problem: As API calls to the old service are propagated to the new service, a situation could arise where data has already been

maintained by directly accessing the new service. Any subsequent call to the old API would then result in overwriting the data stored by the new service as well.

Background information: As soon as the data in the old service is no longer required, customers are free to set the tenant to proxy phase. After that, every API call to the old service is redirected to the new service. Note that the returned payload structure still equals the structure that was used by the old service.

4. Deletion Phase: Once the deprecation phase is over, it is time for SAP to shut down everything that originated in the source environment. That is, the endpoints of the old and now outdated services are removed completely.

Related Information

Value Helps
Location

1.1 Migration Services: Location

Overview of supporting services for data migration into SAP BTP.

Overview

The data migration for the legacy services of SAP IoT Application Enablement is handled via a set of dedicated services that SAP provides. These services are powerful, yet not too complex. Thus, data migration can be seen as a scripting process. It also means that there is **no** migration tool available that you could use to migrate by just pressing a button.

> **i** Note
>
> The location migration described here is exclusively focused on standalone Location data managed by the now outdated `Location` service as described in the API reference (see Location). As opposed to that, any location-related data that are managed as part of the payload of business partner entities is not covered by the migration service described here.

Migration Services

The following endpoints are available for migrating legacy data related to locations:

> **i** Note
>
> The application automatically takes care of migrating legacy location data to the new Geolocation service. In addition, it is used for switching the location migration from replication phase to proxy phase.

Endpoint	Description	Methods
`/LocationMigration`	Sets and returns information about the migration status of the legacy location data.	GET POST
`/MappingMessages`	Retrieves mapping messages, for example, information about truncated texts. See the *Field Mapping* section below.	GET

Payload Fields of the LocationMigration Service Method

Field	Description
`<migrate>`	`true`: Retriggers the migration that has already been done automatically (initial data replication). Calling the service with this parameter set to `true` is useful to resynchronize data in case location data has been changed with the old service **after** the initial replication has already taken place.
`<migrationFinalized>`	`true`: Switch migration from replication phase to proxy phase, which makes the geolocation service the leading application.

Example of a result payload:

```
{
                        "migrate":            false,
                        "migrationFinalized": false
                        }
```

Field Mapping

Both the old and the new location service offer an almost identical set of fields used to specify a location. However, the length of almost all the fields of the new service is shorter than it used to be for the old service. Therefore, during migration, it is unavoidable to have field content cut to the length of the receiving field, thus leading to the requirement of manually reworking the migrated data. See the following table of field mappings:

Location Field Mapping

Legacy Field	Length	SAP BTP Field	Length
`streetName`	255	`Street`	40

Legacy Field	Length	SAP BTP Field	Length
houseNumber	10	House	10
cityName	255	City	40
district	255	District	40
postalCode	10	PostalCode	10
country	2	CountryCode	3
countryDescription	60	Country	40
region	3	RegionCode	3
regionDescription	60	Region	40
longitude	n/a	Part of Geometry (first coordinate)	n/a
latitude	n/a	Part of Geometry (second coordinate)	n/a

> **i** Note
>
> The legacy `Location` service only supports geometrical points that can be described by their longitude and latitude, respectively. As opposed to that, the `Geolocation` service does support different geometry types, which must be described by a set of interconnected coordinates. Therefore, migrating a legacy location into a Geolocation only makes sense for geolocations of type "POI" (point of interest).

Further Changes

In addition to the deviations in field lengths listed in the previous section, the location implementations in the source and target environment differ in the following respects:

- Name: In the source environment, locations do not have names. During migration, the location name field of the target environment is automatically set to the following string value: **`SAP_<source location ID>`**
- Description: In the source environment, locations do not have descriptions. During migration, the location description field of the target environment is automatically set to the following string value: **`<source location ID>`**
- Space: The `Geolocation` service in the target environment supports the concept of a space used as a container for individual locations. The `Location` service in the source environment does **not** support such a concept. Therefore, during initial migration, the system creates a new space named `AE_MIGRATION_SPACE` and assigns all locations to that space.
- Unspecified locations: For the `Location` service in the source environment, it is possible to define a location without specifying its coordinates. This is not allowed in the target environment. Therefore, if a location without coordinates is migrated, the system automatically specifies a point with coordinates (0, 0) for the target `Geolocation` object.
- Country, Region: For the `Location` service in the source environment, entries into these two fields are validated against dedicated database tables that contain all the allowed values. For the `Geolocation` service in the target environment, no such validation is in place. That is, the `Country` and `Region` fields are treated as simple text fields that can contain any string.

- `Etag` handling: In the source environment, write accesses to `Location` objects must be secured via the `If-match` HTTP header field. For `Geolocation` objects in the target environment, this is **not** required.
- Query parameters for collective GET requests: In the source environment, the response payload of collective GET requests to the `Location` service can be modified via the following query parameters:
  - `$orderby`
  - `$filter`
  - `ListFilterID`

  The `Geolocation` service in the target environment does **not** support these parameters.
- Scopes in customer-specific role templates: If you have created your own role templates that contain scopes related to the outdated Location service (that is, `<loc>.c/r/u/d`), make sure to add the corresponding scopes for the new Geolocation service as well (that is, `<geolocation>.c/r/u/d`). For reference, you may want to have a look at the `_Thing_Engineer_Odata` role template that is shipped with SAP IoT and fulfills this requirement; see Role Templates and Scopes.

Related Information

Location - resource-only
Geolocation

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

THE BEST RUN SAP