Document title		Document type
dataModelTranslation		**SD**
Date		Version
2025-10-14		**5.1.0**
Author		Status
Tamás Bordi		**DRAFT**
Contact		Page
tbordi@aitia.ai		**1 (9)**

dataModelTranslation

Service Description

Abstract

This document provides service description for the **dataModelTranslation** service.

Document title
dataModelTranslation
Date
2025-10-14

Version
5.1.0
Status
DRAFT
Page
2 (9)

Contents

	Document title		Version
	dataModelTranslation		**5.1.0**
	Date		Status
	2025-10-14		**DRAFT**
			Page
			3 (9)

ARROWHEAD

1 Overview

This document describes the **dataModelTranslation** service, which provides functionality for translating between one or more data model pairs. A data model is expected to be referenced by a unique ID, and it defines both a specific format (e.g., JSON, XML) and its associated semantics. To enable other systems to consume this service, it needs to be offered through the ServiceRegistry and the handled data model IDs have to be included as part of the service metadata.

The **dataModelTranslation** service contains the following operations:

- *init-translation* accepts the data payload that needs to be translated, creates an internal task for it and immediately returns a unique task ID;

- *get-translation-result* returns the actual status of a given task and the translated data payload (or an error message) if the task has already finished;

- *abort-translation* terminates and removes a given task;

The rest of this document is organized as follows. In Section 2, we describe the abstract message operations provided by the service. In Section 3, we end the document by presenting the data types used by the mentioned operations.

1.1 How This Service Is Meant to Be Used

The data model translation process occurs asynchronously. Service consumers have to first call the *init-translation* operation with the data payload to be translated. If the translation task is successfully initialized, the operation returns a unique task identifier. Using the received task ID, the *get-translation-result* operation has to be polled at a chosen frequency. Until the translation task is completed, the operation returns only the current status; once the task is finished, it returns the translated data payload or an error message if the task has been failed.

1.2 Important Delimitations

Data translation is only possible between data models and in the directions that are defined in the service metadata.

1.3 Access policy

Available for anyone within the Local Cloud with proper permissions.

Document title
dataModelTranslation
Date
2025-10-14

Version
5.1.0
Status
DRAFT
Page
4 (9)

2 Service Operations

This section describes the abstract signatures of each operations of the service. The **dataModelTranslation** service is used to initialize, check and abort a translation task. In particular, each subsection names an operation, an input type and one or two output types (unsuccessful operations can return different structure), in that order. The input type is named inside parentheses, while the output type is preceded by a colon. If the operation has two output types, they are separated by a slash. Input and output types are only denoted when accepted or returned, respectively, by the operation in question. All abstract data types named in this section are defined in Section 3.

2.1 operation init-translation (DataModelTranslationInitRequest) : DataModelTranslationInitResponse / ErrorResponse

Operation *init-translation* returns a unique task ID when the translation task has been successfully initialized. The operation returns immediately and the translation task is executed asynchronously.

2.2 operation get-translation-result (TaskIDRequest) : DataModelTranslationResultResponse / ErrorResponse

Operation *get-translation-result* returns the actual status of a given translation task, the translated data payload if the task has completed successfully, or an error message if the task has failed.

2.3 operation abort-translation (TaskIDRequest) : OperationStatus / ErrorResponse

Operation *abort-translation* terminates and removes a given translation task.

Document title
dataModelTranslation
Date
2025-10-14

Version
5.1.0
Status
DRAFT
Page
5 (9)

3   Information Model

Here, all data objects that can be part of the **dataModelTranslation** service are listed and must be respected by the hosting system. Note that each subsection, which describes one type of object, begins with the *struct* keyword, which is used to denote a collection of named fields, each with its own data type. As a complement to the explicitly defined types in this section, there is also a list of implicit primitive types in Section 3.8, which are used to represent things like hashes and identifiers.

3.1   struct DataModelTranslationInitRequest

Field	Type	Mandatory	Description
inputModelId	DataModelID	yes	The identifier of the data model to be translated.
outputModelId	DataModelID	yes	The identifier of the required result data model.
payload	Payload	yes	The input data to be translated.
settings	Map	no	Any configuration data that might be required by a translation process.

3.2   struct Payload

A byte array encoded with Base64 string (using UTF-8 character set).

3.3   struct Map

An Object which maps String keys to primitive, Object or list values.

3.4   struct DataModelTranslationInitResponse

Field	Type	Description
status	OperationStatus	Status of the operation.
taskId	TaskID	Identifier of the initialized translation task.

Document title
dataModelTranslation
Date
2025-10-14

Version
5.1.0
Status
DRAFT
Page
6 (9)

3.5   struct **ErrorResponse**

Field	Type	Description
status	OperationStatus	Status of the operation.
errorMessage	String	Description of the error.
errorCode	Number	Numerical code of the error.
type	ErrorType	Type of the error.
origin	String	Origin of the error.

3.6   struct **TaskIDRequest**

Field	Type	Mandatory	Description
taskId	TaskID	yes	Identifier of the translation task.

3.7   struct **DataModelTranslationResultResponse**

Field	Type	Description
operationStatus	OperationStatus	Status of the operation.
status	TaskStatus	Status of the translation task.
result	Payload or String	The result (translated) data payload or an error message.
mimeType	MimeType	The MIME type of the result data payload.

Document title
dataModelTranslation
Date
2025-10-14

Version
5.1.0
Status
DRAFT
Page
7 (9)

3.8 Primitives

Types and structures mentioned throughout this document that are assumed to be available to implementations of this service. The concrete interpretations of each of these types and structures must be provided by any IDD document claiming to implement this service.

Type	Description
DataModelID	A string identifier that is intended to be both human and machine-readable. It also defines the specific format and the associated semantics. Must follow camelCase naming convention.
ErrorType	Any suitable type chosen by the implementor of service.
MimeType	Standardized identifier that defines the nature and format of a data payload.
Number	Decimal number.
Object	Set of primitives and possible further objects.
OperationStatus	Logical, textual or numerical value that indicates whether an operation is a success or a failure. Multiple values can be used for success and error cases to give additional information about the nature of the result.
String	A chain of characters.
TaskID	A unique string identifier that belongs to a translation task.
TaskStatus	Logical, textual or numerical value that indicates whether a translation task is pending, in progress, finished or failed.

Document title
dataModelTranslation
Date
2025-10-14

Version
5.1.0
Status
DRAFT
Page
8 (9)

4 References

Document title
dataModelTranslation
Date
2025-10-14

Version
5.1.0
Status
DRAFT
Page
9 (9)

5   Revision History

5.1   Amendments

No.	Date	Version	Subject of Amendments	Author
1	YYYY-MM-DD	5.1.0		Xxx Yyy

5.2   Quality Assurance

No.	Date	Version	Approved by
1	YYYY-MM-DD	5.1.0	