

# AtlasMap Design Time Service API Reference

Version 1.40.0-SNAPSHOT, 2019-02-04

# Design Time Service

1. Core	1
1.1. Paths	1
1.1.1. List FieldActions	1
1.1.2. Upload Library	1
1.1.3. Remove All Mappings	1
1.1.4. Process Mapping	2
1.1.5. Validate Mapping	3
1.1.6. Get Mapping	3
1.1.7. Create Mapping	4
1.1.8. Update Mapping	5
1.1.9. Remove Mapping	5
1.1.10. List Mappings	6
1.1.11. Ping	6
1.2. Definitions	7
1.2.1. ActionDetail	7
1.2.2. ActionDetails	8
1.2.3. ActionParameter	8
1.2.4. ActionParameters	9
1.2.5. Actions	9
1.2.6. AtlasLibraryLoader	9
1.2.7. AtlasMapping	9
1.2.8. AtlasMappingResult	9
1.2.9. Audit	10
1.2.10. Audits	10
1.2.11. BaseMapping	10
1.2.12. ClassLoader	11
1.2.13. Constant	11
1.2.14. Constants	11
1.2.15. DataSource	11
1.2.16. Field	12
1.2.17. FieldGroup	13
1.2.18. LookupEntry	14
1.2.19. LookupTable	14
1.2.20. LookupTables	15
1.2.21. Mapping	15
1.2.22. Mappings	16
1.2.23. ProcessMappingResponse	16
1.2.24. Properties	16

1.2.25. Property	16
1.2.26. StringMap	17
1.2.27. StringMapEntry	17
1.2.28. TargetDocument	17
1.2.29. Validation	17
1.2.30. Validations	18
2. Java	19
2.1. Paths	19
2.1.1. Inspect Class	19
2.1.2. Inspect Class	19
2.1.3. Generate Maven Classpath	20
2.1.4. Simple	20
2.2. Definitions	21
2.2.1. Actions	21
2.2.2. JavaClass	21
2.2.3. JavaEnumField	23
2.2.4. JavaEnumFields	25
2.2.5. JavaField	25
2.2.6. JavaFields	27
2.2.7. MavenClasspathResponse	27
2.2.8. ModifierList	27
2.2.9. StringList	27
3. JSON	28
3.1. Paths	28
3.1.1. Inspect JSON	28
3.1.2. Inspect JSON via URI	28
3.1.3. Simple	29
3.2. Definitions	29
3.2.1. Actions	29
3.2.2. Field	30
3.2.3. Fields	31
3.2.4. JsonDocument	31
3.2.5. JsonInspectionResponse	31
4. XML	32
4.1. Paths	32
4.1.1. Inspect XML	32
4.1.2. Inspect XML via URI	32
4.1.3. Simple	33
4.2. Definitions	33
4.2.1. Actions	33
4.2.2. Field	33

4.2.3. Fields .....	34
4.2.4. XmlDocument .....	35
4.2.5. XmlInspectionResponse .....	35
4.2.6. XmlNamespace .....	35
4.2.7. XmlNamespaces .....	36

# Chapter 1. Core

## 1.1. Paths

### 1.1.1. List FieldActions

GET /fieldActions

#### Description

Retrieves a list of available field action

#### Responses

HTTP Code	Description	Schema
200	Return a list of field action detail	<a href="#">ActionDetails</a>

#### Produces

- `application/json`

### 1.1.2. Upload Library

PUT /library

#### Description

Upload a Java library archive file

#### Responses

HTTP Code	Description	Schema
200	Library upload successful.	No Content

#### Consumes

- `application/octet-stream`

### 1.1.3. Remove All Mappings

DELETE /mapping/RESET

### Description

Remove all mapping files saved on the server

### Responses

HTTP Code	Description	Schema
200	All mapping files were removed successfully	No Content
204	Unable to remove all mapping files	No Content

### Produces

- `application/json`

## 1.1.4. Process Mapping

PUT /mapping/process

### Description

Process Mapping by feeding input data

### Responses

HTTP Code	Description	Schema
200	Return a mapping result	<a href="#">ProcessMappingResponse</a>
204	Skipped empty mapping execution	No Content

### Consumes

- `application/json`

### Produces

- `application/json`

### 1.1.5. Validate Mapping

PUT /mapping/validate

#### Description

Validate mapping file

#### Responses

HTTP Code	Description	Schema
200	Return a validation result	<a href="#">Validations</a>

#### Consumes

- `application/json`

#### Produces

- `application/json`

### 1.1.6. Get Mapping

GET /mapping/{mappingFormat}/{mappingId}

#### Description

Retrieve a mapping file saved on the server

#### Parameters

Type	Name	Description	Schema
Path	<b>mappingFormat</b> at <i>required</i>	Mapping Format	string
Path	<b>mappingId</b> <i>required</i>	Mapping ID	string

#### Responses

HTTP Code	Description	Schema
200	Return a mapping file content	<a href="#">AtlasMapping</a>
204	Mapping file was not found	No Content
500	Mapping file access error	No Content

### Produces

- `application/json`
- `application/xml`
- `application/octet-stream`

## 1.1.7. Create Mapping

```
PUT /mapping/{mappingFormat}/{mappingId}
```

### Description

Save a mapping file on the server

### Parameters

Type	Name	Description	Schema
Path	<b>mappingFormat</b> <i>required</i>	Mapping Format	string
Path	<b>mappingId</b> <i>required</i>	Mapping ID	string

### Responses

HTTP Code	Description	Schema
200	Succeeded	No Content
500	Mapping file save error	No Content

### Consumes

- `application/json`



- `application/xml`
- `application/octet-stream`

#### Produces

- `application/json`

### 1.1.8. Update Mapping

```
POST /mapping/{mappingId}
```

#### Description

Update existing mapping file on the server

#### Parameters

Type	Name	Description	Schema
Path	<b>mappingId</b> <i>required</i>	Mapping ID	string

#### Responses

HTTP Code	Description	Schema
200	Succeeded	No Content

#### Consumes

- `application/json`

#### Produces

- `application/json`

### 1.1.9. Remove Mapping

```
DELETE /mapping/{mappingId}
```

#### Description

Remove a mapping file saved on the server

#### Parameters

Type	Name	Description	Schema
Path	<b>mappingId</b> <i>required</i>	Mapping ID	string

## Responses

HTTP Code	Description	Schema
200	Specified mapping file was removed successfully	No Content
204	Mapping file was not found	No Content

## Produces

- `application/json`

### 1.1.10. List Mappings

GET /mappings

## Description

Retrieves a list of mapping file name saved on the server

## Parameters

Type	Name	Schema
Query	<b>filter</b> <i>optional</i>	string

## Responses

HTTP Code	Description	Schema
200	Return a list of a pair of mapping file name and content	<a href="#">StringMap</a>

## Produces

- `application/json`

### 1.1.11. Ping

GET /ping

## Description

Simple liveness check method used in liveness checks. Must not be protected via authentication.

## Responses

HTTP Code	Description	Schema
200	Return 'pong'	string

# 1.2. Definitions

## 1.2.1. ActionDetail

Name	Schema
<b>className</b> <i>optional</i>	string
<b>custom</b> <i>optional</i>	boolean
<b>method</b> <i>optional</i>	string
<b>name</b> <i>optional</i>	string
<b>parameters</b> <i>optional</i>	<a href="#">ActionParameters</a>
<b>sourceCollectionType</b> <i>optional</i>	enum (ALL, ARRAY, LIST, MAP, NONE)
<b>sourceType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)

Name	Schema
<b>targetCollectionType</b> <i>optional</i>	enum (ALL, ARRAY, LIST, MAP, NONE)
<b>targetType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)

### 1.2.2. ActionDetails

Name	Schema
<b>actionDetail</b> <i>optional</i>	< <a href="#">ActionDetail</a> > array

### 1.2.3. ActionParameter

Name	Schema
<b>description</b> <i>optional</i>	string
<b>displayName</b> <i>optional</i>	string
<b>fieldType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>name</b> <i>optional</i>	string
<b>values</b> <i>optional</i>	< string > array

### 1.2.4. ActionParameters

Name	Schema
<b>parameter</b> <i>optional</i>	< <a href="#">ActionParameter</a> > array

### 1.2.5. Actions

Type : object

### 1.2.6. AtlasLibraryLoader

Name	Schema
<b>empty</b> <i>optional</i>	boolean
<b>parent</b> <i>optional</i>	<a href="#">ClassLoader</a>

### 1.2.7. AtlasMapping

Name	Schema
<b>constants</b> <i>optional</i>	<a href="#">Constants</a>
<b>dataSource</b> <i>required</i>	< <a href="#">DataSource</a> > array
<b>lookupTables</b> <i>optional</i>	<a href="#">LookupTables</a>
<b>mappings</b> <i>required</i>	<a href="#">Mappings</a>
<b>name</b> <i>optional</i>	string
<b>properties</b> <i>optional</i>	<a href="#">Properties</a>

### 1.2.8. AtlasMappingResult

Name	Schema
<b>audits</b> <i>optional</i>	<a href="#">Audits</a>
<b>targetDocuments</b> <i>optional</i>	< <a href="#">TargetDocument</a> > array

### 1.2.9. Audit

Name	Schema
<b>docId</b> <i>optional</i>	string
<b>message</b> <i>optional</i>	string
<b>path</b> <i>optional</i>	string
<b>status</b> <i>optional</i>	enum (ALL, INFO, WARN, ERROR, NONE)
<b>value</b> <i>optional</i>	string

### 1.2.10. Audits

Name	Schema
<b>audit</b> <i>optional</i>	< <a href="#">Audit</a> > array

### 1.2.11. BaseMapping

Name	Schema
<b>alias</b> <i>optional</i>	string
<b>description</b> <i>optional</i>	string

Name	Schema
<b>mappingType</b> <i>optional</i>	enum (ALL, COLLECTION, COMBINE, LOOKUP, MAP, SEPARATE, NONE)

### 1.2.12. ClassLoader

Name	Schema
<b>parent</b> <i>optional</i>	<a href="#">ClassLoader</a>

### 1.2.13. Constant

Name	Schema
<b>fieldType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>name</b> <i>optional</i>	string
<b>value</b> <i>optional</i>	string

### 1.2.14. Constants

Name	Schema
<b>constant</b> <i>optional</i>	< <a href="#">Constant</a> > array

### 1.2.15. DataSource

Name	Schema
<b>dataSourceType</b> <i>optional</i>	enum (SOURCE, TARGET)

Name	Schema
<b>id</b> <i>optional</i>	string
<b>uri</b> <i>optional</i>	string

### 1.2.16. Field

Name	Schema
<b>actions</b> <i>optional</i>	<a href="#">Actions</a>
<b>arrayDimensions</b> <i>optional</i>	integer (int32)
<b>arraySize</b> <i>optional</i>	integer (int32)
<b>collectionType</b> <i>optional</i>	enum (ALL, ARRAY, LIST, MAP, NONE)
<b>docId</b> <i>optional</i>	string
<b>fieldType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>format</b> <i>optional</i>	string
<b>index</b> <i>optional</i>	integer (int32)
<b>path</b> <i>optional</i>	string



Name	Schema
<b>required</b> <i>optional</i>	boolean
<b>status</b> <i>optional</i>	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
<b>value</b> <i>optional</i>	object

### 1.2.17. FieldGroup

Name	Schema
<b>actions</b> <i>optional</i>	<a href="#">Actions</a>
<b>arrayDimensions</b> <i>optional</i>	integer (int32)
<b>arraySize</b> <i>optional</i>	integer (int32)
<b>collectionType</b> <i>optional</i>	enum (ALL, ARRAY, LIST, MAP, NONE)
<b>docId</b> <i>optional</i>	string
<b>field</b> <i>optional</i>	< <a href="#">Field</a> > array
<b>fieldType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>format</b> <i>optional</i>	string

Name	Schema
<b>index</b> <i>optional</i>	integer (int32)
<b>path</b> <i>optional</i>	string
<b>required</b> <i>optional</i>	boolean
<b>status</b> <i>optional</i>	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
<b>value</b> <i>optional</i>	object

### 1.2.18. LookupEntry

Name	Schema
<b>sourceType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>sourceValue</b> <i>optional</i>	string
<b>targetType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>targetValue</b> <i>optional</i>	string

### 1.2.19. LookupTable

Name	Schema
<b>description</b> <i>optional</i>	string
<b>lookupEntry</b> <i>optional</i>	< <a href="#">LookupEntry</a> > array
<b>name</b> <i>optional</i>	string

### 1.2.20. LookupTables

Name	Schema
<b>lookupTable</b> <i>optional</i>	< <a href="#">LookupTable</a> > array

### 1.2.21. Mapping

Name	Schema
<b>alias</b> <i>optional</i>	string
<b>delimiter</b> <i>optional</i>	string
<b>delimiterString</b> <i>optional</i>	string
<b>description</b> <i>optional</i>	string
<b>id</b> <i>optional</i>	string
<b>inputField</b> <i>optional</i>	< <a href="#">Field</a> > array
<b>inputFieldGroup</b> <i>optional</i>	<a href="#">FieldGroup</a>
<b>lookupTableName</b> <i>optional</i>	string

Name	Schema
<b>mappingType</b> <i>optional</i>	enum (ALL, COLLECTION, COMBINE, LOOKUP, MAP, SEPARATE, NONE)
<b>outputField</b> <i>optional</i>	< <a href="#">Field</a> > array
<b>strategy</b> <i>optional</i>	string
<b>strategyClassName</b> <i>optional</i>	string

### 1.2.22. Mappings

Name	Schema
<b>mapping</b> <i>optional</i>	< <a href="#">BaseMapping</a> > array

### 1.2.23. ProcessMappingResponse

Name	Schema
<b>atlasMappingResult</b> <i>optional</i>	<a href="#">AtlasMappingResult</a>
<b>audits</b> <i>optional</i>	<a href="#">Audits</a>
<b>mapping</b> <i>optional</i>	<a href="#">Mapping</a>

### 1.2.24. Properties

Name	Schema
<b>property</b> <i>optional</i>	< <a href="#">Property</a> > array

### 1.2.25. Property

Name	Schema
<b>fieldType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>name</b> <i>optional</i>	string
<b>value</b> <i>optional</i>	string

### 1.2.26. StringMap

Name	Schema
<b>stringMapEntry</b> <i>optional</i>	< <a href="#">StringMapEntry</a> > array

### 1.2.27. StringMapEntry

Name	Schema
<b>name</b> <i>optional</i>	string
<b>value</b> <i>optional</i>	string

### 1.2.28. TargetDocument

Name	Schema
<b>body</b> <i>optional</i>	string
<b>docId</b> <i>optional</i>	string

### 1.2.29. Validation

Name	Schema
<b>id</b> <i>optional</i>	string
<b>message</b> <i>optional</i>	string
<b>scope</b> <i>optional</i>	enum (ALL, DATA_SOURCE, MAPPING, LOOKUP_TABLE, CONSTANT, PROPERTY)
<b>status</b> <i>optional</i>	enum (ALL, INFO, WARN, ERROR, NONE)

### 1.2.30. Validations

Name	Schema
<b>validation</b> <i>optional</i>	< <a href="#">Validation</a> > array

# Chapter 2. Java

## 2.1. Paths

### 2.1.1. Inspect Class

POST /java/class

#### Description

Inspect a Java Class with specified fully qualified class name and return a Document object

#### Responses

HTTP Code	Description	Schema
200	Return a Document object represented by JavaClass	<a href="#">JavaClass</a>

#### Consumes

- `application/json`

#### Produces

- `application/json`

### 2.1.2. Inspect Class

GET /java/class

#### Description

Inspect a Java Class with specified fully qualified class name and return a Document object

#### Parameters

Type	Name	Description	Schema
Query	<b>className</b> <i>optional</i>	The fully qualified class name to inspect	string

#### Responses

HTTP Code	Description	Schema
200	Return a Document object represented by JavaClass	<a href="#">JavaClass</a>

#### Produces

- `application/json`

### 2.1.3. Generate Maven Classpath

POST /java/mavenclasspath

#### Description

Retrieve a maven classpath string

#### Responses

HTTP Code	Description	Schema
200	Return a MavenClasspathResponse object which contains classpath string	<a href="#">MavenClasspathResponse</a>

#### Consumes

- `application/json`

#### Produces

- `application/json`

### 2.1.4. Simple

GET /java/simple

#### Description

Simple hello service

#### Parameters

Type	Name	Description	Schema
Query	<b>from</b> <i>optional</i>	From	string



## Responses

HTTP Code	Description	Schema
200	Return a response	string

## Produces

- `text/plain`

## 2.2. Definitions

### 2.2.1. Actions

Type : object

### 2.2.2. JavaClass

Name	Schema
<b>actions</b> <i>optional</i>	<a href="#">Actions</a>
<b>anonymous</b> <i>optional</i>	boolean
<b>annotation</b> <i>optional</i>	boolean
<b>annotations</b> <i>optional</i>	<a href="#">StringList</a>
<b>arrayDimensions</b> <i>optional</i>	integer (int32)
<b>arraySize</b> <i>optional</i>	integer (int32)
<b>canonicalClassName</b> <i>optional</i>	string
<b>className</b> <i>optional</i>	string

Name	Schema
<b>collectionClassName</b> <i>optional</i>	string
<b>collectionType</b> <i>optional</i>	enum (ALL, ARRAY, LIST, MAP, NONE)
<b>docId</b> <i>optional</i>	string
<b>enumeration</b> <i>optional</i>	boolean
<b>fieldType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>format</b> <i>optional</i>	string
<b>getMethod</b> <i>optional</i>	string
<b>index</b> <i>optional</i>	integer (int32)
<b>interface</b> <i>optional</i>	boolean
<b>javaEnumFields</b> <i>required</i>	<a href="#">JavaEnumFields</a>
<b>javaFields</b> <i>required</i>	<a href="#">JavaFields</a>
<b>localClass</b> <i>optional</i>	boolean
<b>memberClass</b> <i>optional</i>	boolean

Name	Schema
<b>modifiers</b> <i>required</i>	<a href="#">ModifierList</a>
<b>name</b> <i>optional</i>	string
<b>packageName</b> <i>optional</i>	string
<b>parameterizedTypes</b> <i>optional</i>	<a href="#">StringList</a>
<b>path</b> <i>optional</i>	string
<b>primitive</b> <i>optional</i>	boolean
<b>required</b> <i>optional</i>	boolean
<b>setMethod</b> <i>optional</i>	string
<b>status</b> <i>optional</i>	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
<b>synthetic</b> <i>optional</i>	boolean
<b>uri</b> <i>optional</i>	string
<b>value</b> <i>optional</i>	object

### 2.2.3. JavaEnumField

Name	Schema
<b>actions</b> <i>optional</i>	<a href="#">Actions</a>

Name	Schema
<b>arrayDimensions</b> <i>optional</i>	integer (int32)
<b>arraySize</b> <i>optional</i>	integer (int32)
<b>className</b> <i>optional</i>	string
<b>collectionType</b> <i>optional</i>	enum (ALL, ARRAY, LIST, MAP, NONE)
<b>docId</b> <i>optional</i>	string
<b>fieldType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>format</b> <i>optional</i>	string
<b>index</b> <i>optional</i>	integer (int32)
<b>name</b> <i>optional</i>	string
<b>ordinal</b> <i>optional</i>	integer (int32)
<b>path</b> <i>optional</i>	string
<b>required</b> <i>optional</i>	boolean
<b>status</b> <i>optional</i>	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)

Name	Schema
<b>value</b> <i>optional</i>	object

#### 2.2.4. JavaEnumFields

Name	Schema
<b>javaEnumField</b> <i>optional</i>	< <a href="#">JavaEnumField</a> > array

#### 2.2.5. JavaField

Name	Schema
<b>actions</b> <i>optional</i>	<a href="#">Actions</a>
<b>annotations</b> <i>optional</i>	<a href="#">StringList</a>
<b>arrayDimensions</b> <i>optional</i>	integer (int32)
<b>arraySize</b> <i>optional</i>	integer (int32)
<b>canonicalClassName</b> <i>optional</i>	string
<b>className</b> <i>optional</i>	string
<b>collectionClassName</b> <i>optional</i>	string
<b>collectionType</b> <i>optional</i>	enum (ALL, ARRAY, LIST, MAP, NONE)
<b>docId</b> <i>optional</i>	string

Name	Schema
<b>fieldType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>format</b> <i>optional</i>	string
<b>getMethod</b> <i>optional</i>	string
<b>index</b> <i>optional</i>	integer (int32)
<b>modifiers</b> <i>required</i>	<a href="#">ModifierList</a>
<b>name</b> <i>optional</i>	string
<b>parameterizedTypes</b> <i>optional</i>	<a href="#">StringList</a>
<b>path</b> <i>optional</i>	string
<b>primitive</b> <i>optional</i>	boolean
<b>required</b> <i>optional</i>	boolean
<b>setMethod</b> <i>optional</i>	string
<b>status</b> <i>optional</i>	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
<b>synthetic</b> <i>optional</i>	boolean

Name	Schema
<b>value</b> <i>optional</i>	object

### 2.2.6. JavaFields

Name	Schema
<b>javaField</b> <i>optional</i>	< <a href="#">JavaField</a> > array

### 2.2.7. MavenClasspathResponse

Name	Schema
<b>classpath</b> <i>optional</i>	string
<b>errorMessage</b> <i>optional</i>	string
<b>executionTime</b> <i>optional</i>	integer (int64)

### 2.2.8. ModifierList

Name	Schema
<b>modifier</b> <i>optional</i>	< enum (ALL, ABSTRACT, FINAL, INTERFACE, NATIVE, PACKAGE_PRIVATE, PUBLIC, PROTECTED, PRIVATE, STATIC, STRICT, SYNCHRONIZED, TRANSIENT, VOLATILE, NONE) > array

### 2.2.9. StringList

Name	Schema
<b>string</b> <i>optional</i>	< string > array

# Chapter 3. JSON

## 3.1. Paths

### 3.1.1. Inspect JSON

POST /json/inspect

#### Description

Inspect a JSON schema or instance and return a Document object

#### Responses

HTTP Code	Description	Schema
200	Return a Document object represented by JsonDocument	<a href="#">JsonInspectionResponse</a>

#### Consumes

- `application/json`

#### Produces

- `application/json`

### 3.1.2. Inspect JSON via URI

GET /json/inspect

#### Description

**NOT IMPLEMENTED** Inspect a JSON schema or instance located at specified URI and return a Document object

#### Parameters

Type	Name	Description	Schema
Query	<code>type</code> <i>optional</i>		string



Type	Name	Description	Schema
Query	<b>uri</b> <i>optional</i>	URI for JSON schema or instance	string

## Responses

HTTP Code	Description	Schema
200	Return a Document object represented by JsonDocument	<a href="#">JsonDocument</a>

## Produces

- `application/json`

### 3.1.3. Simple

```
GET /json/simple
```

## Description

Simple hello service

## Parameters

Type	Name	Description	Schema
Query	<b>from</b> <i>optional</i>	From	string

## Responses

HTTP Code	Description	Schema
200	Return a response	string

## Produces

- `text/plain`

## 3.2. Definitions

### 3.2.1. Actions

Type : object

### 3.2.2. Field

Name	Schema
<b>actions</b> <i>optional</i>	<a href="#">Actions</a>
<b>arrayDimensions</b> <i>optional</i>	integer (int32)
<b>arraySize</b> <i>optional</i>	integer (int32)
<b>collectionType</b> <i>optional</i>	enum (ALL, ARRAY, LIST, MAP, NONE)
<b>docId</b> <i>optional</i>	string
<b>fieldType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>format</b> <i>optional</i>	string
<b>index</b> <i>optional</i>	integer (int32)
<b>path</b> <i>optional</i>	string
<b>required</b> <i>optional</i>	boolean
<b>status</b> <i>optional</i>	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
<b>value</b> <i>optional</i>	object

### 3.2.3. Fields

Name	Schema
<b>field</b> <i>optional</i>	< <a href="#">Field</a> > array

### 3.2.4. JsonDocument

Name	Schema
<b>fields</b> <i>required</i>	<a href="#">Fields</a>

### 3.2.5. JsonInspectionResponse

Name	Schema
<b>errorMessage</b> <i>optional</i>	string
<b>executionTime</b> <i>optional</i>	integer (int64)
<b>jsonDocument</b> <i>optional</i>	<a href="#">JsonDocument</a>

# Chapter 4. XML

## 4.1. Paths

### 4.1.1. Inspect XML

POST /xml/inspect

#### Description

Inspect a XML schema or instance and return a Document object

#### Responses

HTTP Code	Description	Schema
200	Return a Document object represented by XmlDocument	<a href="#">XmlInspectionResponse</a>

#### Consumes

- `application/json`

#### Produces

- `application/json`

### 4.1.2. Inspect XML via URI

GET /xml/inspect

#### Description

Inspect a XML schema or instance located at specified URI and return a Document object

#### Parameters

Type	Name	Schema
Query	<b>type</b> <i>optional</i>	string
Query	<b>uri</b> <i>optional</i>	string

## Responses

HTTP Code	Description	Schema
200	Return a Document object represented by XmlDocument	<a href="#">XmlDocument</a>

## Produces

- `application/json`

### 4.1.3. Simple

GET /xml/simple

## Description

Simple hello service

## Parameters

Type	Name	Schema
Query	<b>from</b> <i>optional</i>	string

## Responses

HTTP Code	Description	Schema
200	Return a response	string

## Produces

- `text/plain`

## 4.2. Definitions

### 4.2.1. Actions

Type : object

### 4.2.2. Field

Name	Schema
<b>actions</b> <i>optional</i>	<a href="#">Actions</a>
<b>arrayDimensions</b> <i>optional</i>	integer (int32)
<b>arraySize</b> <i>optional</i>	integer (int32)
<b>collectionType</b> <i>optional</i>	enum (ALL, ARRAY, LIST, MAP, NONE)
<b>docId</b> <i>optional</i>	string
<b>fieldType</b> <i>optional</i>	enum (ANY, ANY_DATE, BIG_INTEGER, BOOLEAN, BYTE, BYTE_ARRAY, CHAR, COMPLEX, DATE, DATE_TIME, DATE_TIME_TZ, DATE_TZ, DECIMAL, DOUBLE, FLOAT, INTEGER, LONG, NONE, NUMBER, SHORT, STRING, TIME, TIME_TZ, UNSIGNED_BYTE, UNSIGNED_INTEGER, UNSIGNED_LONG, UNSIGNED_SHORT, UNSUPPORTED)
<b>format</b> <i>optional</i>	string
<b>index</b> <i>optional</i>	integer (int32)
<b>path</b> <i>optional</i>	string
<b>required</b> <i>optional</i>	boolean
<b>status</b> <i>optional</i>	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
<b>value</b> <i>optional</i>	object

### 4.2.3. Fields

Name	Schema
<b>field</b> <i>optional</i>	< <a href="#">Field</a> > array

#### 4.2.4. XmlDocument

Name	Schema
<b>fields</b> <i>required</i>	<a href="#">Fields</a>
<b>xmlNamespaces</b> <i>required</i>	<a href="#">XmlNamespaces</a>

#### 4.2.5. XmlInspectionResponse

Name	Schema
<b>errorMessage</b> <i>optional</i>	string
<b>executionTime</b> <i>optional</i>	integer (int64)
<b>xmlDocument</b> <i>optional</i>	<a href="#">XmlDocument</a>

#### 4.2.6. XmlNamespace

Name	Schema
<b>alias</b> <i>optional</i>	string
<b>locationUri</b> <i>optional</i>	string
<b>targetNamespace</b> <i>optional</i>	boolean
<b>uri</b> <i>optional</i>	string

### 4.2.7. XmlNamespaces

Name	Schema
<b>xmlNamespace</b> <i>optional</i>	< <a href="#">XmlNamespace</a> > array