AtlasMap Design Time Service API Reference

Version 1.40.0-SNAPSHOT, 2019-02-05

Design Time Service

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	ActionDetails

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	ProcessMappingResponse
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	Validations

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	**mappingFormat** *required*	Mapping Format	string
Path	**mappingId** *required*	Mapping ID	string

Responses

HTTP Code	Description	Schema
200	Return a mapping file content	AtlasMapping
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	**mappingFormat** *required*	Mapping Format	string
Path	**mappingId** *required*	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	**mappingId** *required*	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	**mappingId** *required*	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	**filter** *optional*	string

Responses

HTTP Code	Description	Schema
200	Return a list of a pair of mapping file name and content	StringMap

Produces

- `application/json`

1.1.11. Ping

```
GET /ping
```

Description

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

Responses

HTTP Code	Description	Schema
200	Return 'pong'	string

1.2. Definitions

1.2.1. ActionDetail

Name	Schema
className *optional*	string
custom *optional*	boolean
method *optional*	string
name *optional*	string
parameters *optional*	ActionParameters
sourceCollectionType *optional*	enum (ALL, ARRAY, LIST, MAP, NONE)
sourceType *optional*	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
targetCollectionType *optional*	enum (ALL, ARRAY, LIST, MAP, NONE)
targetType *optional*	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
actionDetail *optional*	< ActionDetail > array

1.2.3. ActionParameter

Name	Schema
description *optional*	string
displayName *optional*	string
fieldType *optional*	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 *optional*	string
values *optional*	< string > array

1.2.4. ActionParameters

Name	Schema
parameter *optional*	< ActionParameter > array

1.2.5. Actions

Type : object

1.2.6. AtlasLibraryLoader

Name	Schema
empty *optional*	boolean
parent *optional*	ClassLoader

1.2.7. AtlasMapping

Name	Schema
constants *optional*	Constants
dataSource *required*	< DataSource > array
lookupTables *optional*	LookupTables
mappings *required*	Mappings
name *optional*	string
properties *optional*	Properties

1.2.8. AtlasMappingResult

Name	Schema
audits *optional*	Audits
targetDocuments *optional*	< TargetDocument > array

1.2.9. Audit

Name	Schema
docId *optional*	string
message *optional*	string
path *optional*	string
status *optional*	enum (ALL, INFO, WARN, ERROR, NONE)
value *optional*	string

1.2.10. Audits

Name	Schema
audit *optional*	< Audit > array

1.2.11. BaseMapping

Name	Schema
alias *optional*	string
description *optional*	string

Name	Schema
mappingType *optional*	enum (ALL, COLLECTION, COMBINE, LOOKUP, MAP, SEPARATE, NONE)

1.2.12. ClassLoader

Name	Schema
parent *optional*	ClassLoader

1.2.13. Constant

Name	Schema
fieldType *optional*	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 *optional*	string
value *optional*	string

1.2.14. Constants

Name	Schema
constant *optional*	< Constant > array

1.2.15. DataSource

Name	Schema
dataSourceType *optional*	enum (SOURCE, TARGET)

Name	Schema
id *optional*	string
uri *optional*	string

1.2.16. Field

Name	Schema
actions *optional*	Actions
arrayDimensions *optional*	integer (int32)
arraySize *optional*	integer (int32)
collectionType *optional*	enum (ALL, ARRAY, LIST, MAP, NONE)
docId *optional*	string
fieldType *optional*	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)
format *optional*	string
index *optional*	integer (int32)
path *optional*	string

Name	Schema
required *optional*	boolean
status *optional*	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
value *optional*	object

1.2.17. FieldGroup

Name	Schema
actions *optional*	Actions
arrayDimensions *optional*	integer (int32)
arraySize *optional*	integer (int32)
collectionType *optional*	enum (ALL, ARRAY, LIST, MAP, NONE)
docId *optional*	string
field *optional*	< Field > array
fieldType *optional*	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)
format *optional*	string

Name	Schema
index *optional*	integer (int32)
path *optional*	string
required *optional*	boolean
status *optional*	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
value *optional*	object

1.2.18. LookupEntry

Name	Schema
sourceType *optional*	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)
sourceValue *optional*	string
targetType *optional*	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)
targetValue *optional*	string

1.2.19. LookupTable

Name	Schema
description *optional*	string
lookupEntry *optional*	< LookupEntry > array
name *optional*	string

1.2.20. LookupTables

Name	Schema
lookupTable *optional*	< LookupTable > array

1.2.21. Mapping

Name	Schema
alias *optional*	string
delimiter *optional*	string
delimiterString *optional*	string
description *optional*	string
id *optional*	string
inputField *optional*	< Field > array
inputFieldGroup *optional*	FieldGroup
lookupTableName *optional*	string

Name	Schema
mappingType *optional*	enum (ALL, COLLECTION, COMBINE, LOOKUP, MAP, SEPARATE, NONE)
outputField *optional*	< Field > array
strategy *optional*	string
strategyClassName *optional*	string

1.2.22. Mappings

Name	Schema
mapping *optional*	< BaseMapping > array

1.2.23. ProcessMappingResponse

Name	Schema
atlasMappingResult *optional*	AtlasMappingResult
audits *optional*	Audits
mapping *optional*	Mapping

1.2.24. Properties

Name	Schema
property *optional*	< Property > array

1.2.25. Property

Name	Schema
fieldType *optional*	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 *optional*	string
value *optional*	string

1.2.26. StringMap

Name	Schema
stringMapEntry *optional*	< StringMapEntry > array

1.2.27. StringMapEntry

Name	Schema
name *optional*	string
value *optional*	string

1.2.28. TargetDocument

Name	Schema
body *optional*	string
docId *optional*	string

1.2.29. Validation

Name	Schema
id *optional*	string
message *optional*	string
scope *optional*	enum (ALL, DATA_SOURCE, MAPPING, LOOKUP_TABLE, CONSTANT, PROPERTY)
status *optional*	enum (ALL, INFO, WARN, ERROR, NONE)

1.2.30. Validations

Name	Schema
validation *optional*	< Validation > 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	JavaClass

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	**className** *optional*	The fully qualified class name to inspect	string

Responses

HTTP Code	Description	Schema
200	Return a Document object represented by JavaClass	JavaClass

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	MavenClasspathResponse

Consumes

- `application/json`

Produces

- `application/json`

2.1.4. Simple

```
GET /java/simple
```

Description

Simple hello service

Parameters

Type	Name	Description	Schema
Query	**from** *optional*	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
actions *optional*	Actions
annonymous *optional*	boolean
annotation *optional*	boolean
annotations *optional*	StringList
arrayDimensions *optional*	integer (int32)
arraySize *optional*	integer (int32)
canonicalClassName *optional*	string
className *optional*	string

Name	Schema
collectionClassName *optional*	string
collectionType *optional*	enum (ALL, ARRAY, LIST, MAP, NONE)
docId *optional*	string
enumeration *optional*	boolean
fieldType *optional*	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)
format *optional*	string
getMethod *optional*	string
index *optional*	integer (int32)
interface *optional*	boolean
javaEnumFields *required*	JavaEnumFields
javaFields *required*	JavaFields
localClass *optional*	boolean
memberClass *optional*	boolean

Name	Schema
modifiers *required*	ModifierList
name *optional*	string
packageName *optional*	string
parameterizedTypes *optional*	StringList
path *optional*	string
primitive *optional*	boolean
required *optional*	boolean
setMethod *optional*	string
status *optional*	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
synthetic *optional*	boolean
uri *optional*	string
value *optional*	object

2.2.3. JavaEnumField

Name	Schema
actions *optional*	Actions

Name	Schema
arrayDimensions *optional*	integer (int32)
arraySize *optional*	integer (int32)
className *optional*	string
collectionType *optional*	enum (ALL, ARRAY, LIST, MAP, NONE)
docId *optional*	string
fieldType *optional*	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)
format *optional*	string
index *optional*	integer (int32)
name *optional*	string
ordinal *optional*	integer (int32)
path *optional*	string
required *optional*	boolean
status *optional*	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)

Name	Schema
value *optional*	object

2.2.4. JavaEnumFields

Name	Schema
javaEnumField *optional*	< JavaEnumField > array

2.2.5. JavaField

Name	Schema
actions *optional*	Actions
annotations *optional*	StringList
arrayDimensions *optional*	integer (int32)
arraySize *optional*	integer (int32)
canonicalClassName *optional*	string
className *optional*	string
collectionClassName *optional*	string
collectionType *optional*	enum (ALL, ARRAY, LIST, MAP, NONE)
docId *optional*	string

Name	Schema
fieldType *optional*	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)
format *optional*	string
getMethod *optional*	string
index *optional*	integer (int32)
modifiers *required*	ModifierList
name *optional*	string
parameterizedTypes *optional*	StringList
path *optional*	string
primitive *optional*	boolean
required *optional*	boolean
setMethod *optional*	string
status *optional*	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
synthetic *optional*	boolean

Name	Schema
value *optional*	object

2.2.6. JavaFields

Name	Schema
javaField *optional*	< JavaField > array

2.2.7. MavenClasspathResponse

Name	Schema
classpath *optional*	string
errorMessage *optional*	string
executionTime *optional*	integer (int64)

2.2.8. ModifierList

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

2.2.9. StringList

Name	Schema
string *optional*	< 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	JsonInspectionResponse

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	**type** *optional*		string

Type	Name	Description	Schema
Query	**uri** *optional*	URI for JSON schema or instance	string

Responses

HTTP Code	Description	Schema
200	Return a Document object represented by JsonDocument	JsonDocument

Produces

- `application/json`

3.1.3. Simple

```
GET /json/simple
```

Description

Simple hello service

Parameters

Type	Name	Description	Schema
Query	**from** *optional*	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
actions *optional*	Actions
arrayDimensions *optional*	integer (int32)
arraySize *optional*	integer (int32)
collectionType *optional*	enum (ALL, ARRAY, LIST, MAP, NONE)
docId *optional*	string
fieldType *optional*	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)
format *optional*	string
index *optional*	integer (int32)
path *optional*	string
required *optional*	boolean
status *optional*	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
value *optional*	object

3.2.3. Fields

Name	Schema
field *optional*	< Field > array

3.2.4. JsonDocument

Name	Schema
fields *required*	Fields

3.2.5. JsonInspectionResponse

Name	Schema
errorMessage *optional*	string
executionTime *optional*	integer (int64)
jsonDocument *optional*	JsonDocument

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	XmlInspectionResponse

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	**type** *optional*	string
Query	**uri** *optional*	string

Responses

HTTP Code	Description	Schema
200	Return a Document object represented by XmlDocument	XmlDocument

Produces

- `application/json`

4.1.3. Simple

```
GET /xml/simple
```

Description

Simple hello service

Parameters

Type	Name	Schema
Query	**from** *optional*	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
actions *optional*	Actions
arrayDimensions *optional*	integer (int32)
arraySize *optional*	integer (int32)
collectionType *optional*	enum (ALL, ARRAY, LIST, MAP, NONE)
docId *optional*	string
fieldType *optional*	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)
format *optional*	string
index *optional*	integer (int32)
path *optional*	string
required *optional*	boolean
status *optional*	enum (SUPPORTED, UNSUPPORTED, CACHED, ERROR, NOT_FOUND, BLACK_LIST)
value *optional*	object

4.2.3. Fields

Name	Schema
field *optional*	< Field > array

4.2.4. XmlDocument

Name	Schema
fields *required*	Fields
xmlNamespaces *required*	XmlNamespaces

4.2.5. XmlInspectionResponse

Name	Schema
errorMessage *optional*	string
executionTime *optional*	integer (int64)
xmlDocument *optional*	XmlDocument

4.2.6. XmlNamespace

Name	Schema
alias *optional*	string
locationUri *optional*	string
targetNamespace *optional*	boolean
uri *optional*	string

4.2.7. XmlNamespaces

Name	Schema
xmlNamespace *optional*	< XmlNamespace > array