

मसौदा भारतीय मानक
एकीकृत डाटा एक्सचेंज
भाग 4: अनुपालन विशिष्टता
अनुभाग 1: कैटलॉग सेवा

Draft Indian Standard
Unified Data Exchange
Part 4: Compliance Specifications
Section 1: Catalogue Service

ICS 33.020, 35.020

© BIS 2023
BUREAU OF INDIAN STANDARDS
MANAK BHAVAN, 9 BAHADUR SHAH ZAFAR MARG
NEW DELHI 110002

November 2023

Price Group: XXX

FOREWORD

This Draft Indian Standard may be adopted by the Bureau of Indian Standards, after the draft is finalised by the Smart Infrastructure Sectional Committee, and approved by the Electronics and Information Technology Division Council.

DRAFT

Contents

0 INTRODUCTION	5
1 SCOPE	6
2 REFERENCES	6
2.1 Normative References.....	6
2.2 Informative References	7
3 DEFINITIONS	8
3.1 Terminology.....	8
3.2 Abbreviations	9
4 CATALOGUE SERVICE FUNCTIONAL PROFILES	10
5 TEST CASES FOR FUNCTIONAL PROFILES	13
5.1 2XX Test cases for Management APIs	13
5.1.1 2XX Test for Management APIs.....	14
5.2 4XX Test cases for Management APIs	15
5.2.1 4XX Test for Management APIs.....	15
5.3 2XX Test cases for Discovery APIs (G0).....	16
5.3.1 2XX Test for Attribute (Property) Search.....	17
5.3.2 2XX Test for List Search	18
5.3.3 2XX Test for Tag Search.....	19
5.3.4 2XX Test for Count Search	19
5.4 4XX Test cases for discovery (G0)	20
5.4.1 4XX Test for Attribute (Property) Search.....	20
5.4.2 4XX Test for List Search.....	22
5.4.4 4XX Test for Tag Search	22
5.4.5 4XX Test for Count Search	23
5.5 2XX Test cases for Discovery APIs (G1).....	25
5.5.1 2XX for Relationship APIs(G1)	25
5.5.2 2XX for Relationship Search(G1)	26
5.5.3 2XX for Geo-spatial Search(G1)	27

5.5.3.1 2XX for Point(Circle) Search	27
5.5.3.2 2XX for Polygon Search	27
5.5.3.3 2XX for Line String Search.....	28
5.5.3.4 2XX for BBox Search.....	29
5.5.4 2XX for Text Search (G1)	29
5.5.5 2XX for Count APIs (G1)	30
5.6. 4XX Test cases for Discovery(G1).....	31
5.6.1 4XX Tests for Relationship APIs(G1)	31
5.6.2 4XX Tests for Geo-Spatial Search(G1).....	32
5.6.2.1 4XX Tests for Point(Circle) Search(G1).....	32
5.6.2.2 4XX Tests for Polygon Search(G1).....	34
5.6.2.3 4XX Tests for Linestring Search(G1).....	36
5.6.2.1 4XX Tests for Bbox Search(G1).....	38
5.6.3 4XX Tests for Text Search(G1)	40
5.6.4 4XX Tests for Count APIs(G1)	41
5.7. 2XX Test cases for Discovery APIs(G2).....	43
5.7.1 2XX for Complex Search.....	43
5.7.2 2XX Test cases for Response Filtering	44
5.8. 4XX Test Cases for Discovery(G2)	48
5.8.1 4XX Tests for Response filtering (G2)	48
ANNEXURES.....	50
ANNEX-A.....	50
ANNEX-B.....	52
ANNEX-C.....	52
ANNEX-D.....	52

0 INTRODUCTION

The smart cities are generating an enormous amount of data. If harnessed in the right way, this data can empower the stakeholders viz, the providers, the consumers and the governing agencies in solving the key challenges faced by the cities and add value by building innovative applications. One issue faced by the current smart cities is the inability to exchange data efficiently due to the proprietary and ad-hoc nature of the interfaces and their implementations.

To address the data exchange bottlenecks, a Unified Data Exchange (DX) layer, which provides a standardised framework for accessing data in a unified format and allowing authorised data sharing amongst different entities, was defined in the Indian Standard IS 18003 (Part 1) Unified Data Exchange Architecture and the Indian Standard IS 18003 (Part 2) : API Specifications.

The Data Exchange (DX) layer specifies three sets of services, namely the Catalogue Service, the Authorization Service and the Resource Access service. The detailed Application Programming Interface (APIs) specifications for each of these services are provided in - IS 18003 (Part 2): API Specifications. In particular, the Catalogue Service forms the data discovery plane for the DX layer. It defines discovery interfaces to allow data consumers to discover a resource. Further, it defines management interfaces to allow data providers to publish metadata of a resource along with data access information.

This standard IS 18003 (Part 3): Compliance Specifications Sec-2 (catalogue Service) specifies the abstract test suite to define compliance to the catalogue Service as defined in IS 18003 (Part 2): API Specifications. The test suite defines the minimum functionality required for any compliant Catalogue Service implementation.

The compliance specifications are divided into 5 sections. Section 1 gives the scope of this Compliance specification document. Section 2 lists the Normative and Informative references. Section 3 gives the definitions on various terminologies and abbreviations used in this document. Section 4 details the Catalogue Access Service functional profiles and the details of the tests are presented in Section 5.

1 SCOPE

The Indian Standard IS 18003 (Part 1) describes the Unified Data Exchange reference architecture and the Indian standard IS 18003 (Part 2) defines the API specifications for the DX interfaces identified in DX reference architecture. This Indian standard IS18003 (Part 3/Sec 2) defines the compliance test suites for any external implementation of the Catalogue Service for the Data Exchange (DX) as specified in IS18003 (Part 2) of the standard. Compliance specifications for Data Access Service and Authorization Service are available as separate IS18003 (Part 3) standards.

The target audience for this standard are the developers of DX Catalogue Services and the developers of testing and compliance suites belonging to independent testing and certification agencies. This standard will also be helpful for DX consumers to understand the implementation details of the APIs and the functional profiles.

2 REFERENCES

2.1 Normative References

The following referenced documents are necessary for the application of the present document.

IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content". Available at <https://tools.ietf.org/html/rfc7231>

IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests". Available at <https://tools.ietf.org/html/rfc7232>.

IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax". Available at <https://tools.ietf.org/html/rfc3986>.

IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format". Available at <https://tools.ietf.org/html/rfc8259>.

IETF RFC 7946: "The GeoJSON Format". Available at <https://tools.ietf.org/html/rfc7946>.

IETF RFC 8141: "Uniform Resource Names (URNs)". Available at <https://tools.ietf.org/html/rfc8141>.

Open Geospatial Consortium Inc. OGC 06-103r4: "OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture". Available at https://portal.opengeospatial.org/files/?artifact_id=25355.

JSON-LD 1.1: A JSON based serialization for linked data. W3C Recommendation, July 2020. Available at: <https://www.w3.org/TR/json-ld11/>

IETF RFC 7807: Problem Details for HTTP APIs. Available at: <https://tools.ietf.org/html/rfc7807>

ISO 8601: 2004: "Data elements and interchange formats -- Information interchange -- Representation of dates and times". Available at http://www.iso.org/iso/catalogue_detail?csnumber=40874.

IETF RFC 2818: "HTTP Over TLS". Available at <https://tools.ietf.org/html/rfc2818>.

IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2". Available at <https://tools.ietf.org/html/rfc5246>.

IS 18003 (Part 1) : 2020 Unified Data Exchange Framework: Part 1 Architecture, BIS Standard

IS 17203: 2021: Unified Digital Infrastructure: Data Layer Reference Architecture, BIS Standard, Under Publication.

2.2 Informative References

ETSI GS CIM 009 V1.4.1 (2021-02), Context Information Management (CIM); NGSI-LD API. Available at:

https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.04.01_60/gs_CIM009v010401p.pdf

OpenID Connect Core 1.0, Available at https://openid.net/specs/openid-connect-core-1_0.html

IETF RFC 6749: The OAuth 2.0 Authorization Framework. Available at <https://tools.ietf.org/html/rfc6749>

IETF RFC 6749: The OAuth 2.0 Authorization Framework: The Bearer Token Usage. Available at: <https://tools.ietf.org/html/rfc6750>.

ETF RFC 7519: JSON Web Token (JWT). Available at: <https://tools.ietf.org/html/rfc7519>

MQTT 5.0, OASIS Standard. Available at <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>

Advanced Message Queuing Protocol (AMQP): v0.9.1. Available at: <https://www.amqp.org/specification/0-9-1/amqp-org-download>

JSON Schema. Available at: <https://json-schema.org/>

IETF RFC 7396: "JSON Merge Patch". Available at <https://tools.ietf.org/html/rfc7396>.

Linked Data. Tim Berners-Lee. Personal View, imperfect but published. Available at: <http://www.w3.org/DesignIssues/LinkedData.html>

IETF RFC 7522: Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants. Available at: <https://tools.ietf.org/html/rfc7522>

IETF RFC 3987: Internationalized Resource Identifiers (IRIs). Available at: <https://tools.ietf.org/html/rfc3987>

3 DEFINITIONS

3.1 Terminology

Table 3.1: Key terms used throughout this document

Term	Explanation
Provider	Legal Entity: Human (possibly delegated by an Organisation), Organisation or an organisational role that has responsibility to provide authorisation to use resources.
Consumer	Legal Entity: Human or Organisation or an organisational Role that consumes a resource via a web or mobile App.
Data Exchange (DX)	Service: Hosts and manages meta-data about data resources, manages authorisation for accessing the resources and provides data access for the available data resources.
DX Catalogue service	Service: Provides services to manage meta-information about data resources and provides search functionalities to discover data resources hosted with the data exchange. A software entity providing this service will be referred to as Catalogue Server.
DX Resource Access Service	Service: Serves resources to authorised Apps/Consumers. A software entity providing this service will be referred to as Resource Server.
DX Authorization Service	Service: Provides authorization to access data for data resources in accordance to the access policies set for the resources. A software entity providing this service will be referred to as Authorization Server.
Entity	A DX Catalogue Service representation of a DX resource, a DX resource group, a DX Resource Server or a DX Provider.
Authorization Token	A digital entity that is used to present the authorization credentials to the Resource Server.
Authentication Token	A digital entity used to prove the identity of a user to the DX Authorization Service.
Catalogue Item	An entry in the DX Catalogue that describes the meta-information associated with DX entities. Information contained in a catalogue item depends upon the type of the item.
App	Application: Software (like a mobile app, web app, device app or server app), that uses resources to provide a service or experience to the Consumer.

ProviderApp	Application: An App that enables a Provider to manage the meta-data and access control in the data exchange, for the resources they are responsible for.
--------------------	--

3.2 Abbreviations

Table 3.2: List of abbreviations used in this document

Abbreviation	Definition
DX	Data Exchange
JSON	JavaScript Object Notation
API	Application Programming Interface
RS	Resource Server
CS	Catalogue Server
AS	Authorization Server
RG	Resource Group
RI	Resource Item
TLS	Transport Level Security
CRUD	Create, Read, Update, Delete API operations
JSON-LD	JavaScript Object Notation for Linked Data
JWT	JSON Web Token
URN	Uniform Resource Name
URL	Uniform Resource Locator
IRI	Internationalised Resource Identifier
AMQP	Advanced Message Queuing Protocol
MQTT	Message Queuing Telemetry Transport
UUID	Universally Unique Identifier
XML	eXtensible Markup Language

4 CATALOGUE SERVICE FUNCTIONAL PROFILES

Within the scope of this standard, a functional profile defines a logical grouping of functionalities provided by the catalogue service. Different functional profiles are mutually exclusive in terms of functionality coverage and together all the profiles cover the full functionality defined for the catalogue service as defined in IS 18003 (Part 2) of the standard. This specification provides abstract test suites to define compliance for a given functional profile. Thus, the functional profile defines the smallest catalogue service functionality grouping for which compliance shall be provided.

The Catalogue Service provides two functionalities, namely Management and Discovery. Management APIs are meant for providers to Create, Update, Delete or Get items on the catalogue. Discovery APIs provide the functionality of Attribute (Property), List, Tag, Relationship, Spatial, Text and Complex searches with which users can query a metadata of a Resource Item (RI), a Resource Group (RG), a Resource Server (RS) or a Provider in a DX system. The details of the catalogue service functionalities are discussed in section 5 of the IS 18003 (Part 2) of the standard.

The Minimum Viable Product (MVP) requires Management APIs to Create, Update, Delete and Get documents on the Catalogue Server. Having a suite of management capabilities is mandatory before creating a set of Discovery APIs. Table 4.1 lists all the APIs a catalogue service can expose.

Based on the requirements from a MVP point of view, the discovery test cases have been classified into three groups. Group 0 (G0) defines the basic requirements, Group 1 (G1) and Group 2 (G2) defines the advanced requirements which have to be adhered to during implementation. Table 4.2, Table 4.3 and Table 4.4 give the functionality provided by Group 0, Group 1 and Group 2 respectively. It must be noted that the implementer can choose to implement either Group 0 or Group 1 or a possible combination of both (termed as Group 2).

Management APIs	Discovery APIs		
	Group 0	Group 1	Group 2
Create Item	Attribute (Property) Search	Relationship APIs	Complex Search
Update Item	List Search	Geospatial Search	<ol style="list-style-type: none"> 1. Attribute + Geospatial 2. Geospatial + Text 3. Attribute + Text 4. Attribute + Geospatial + Text 5. Tag + Geospatial 6. Tag + Text 7. Tag + Geospatial + Text
Delete Item	Tag Search	<ol style="list-style-type: none"> 1. Point (Circle) Search 2. Polygon Search 3. Linestring Search 4. BBox Search 	
Get Item	Count APIs		
		Text Search	
		Count APIs	

Table 4.1 Functional profiles (Management & Discovery)

Group	Functionality
G0(Attribute)	Search documents based on given attribute(s)
G0(List)	List documents based on a certain type
G0(Tag)	List documents that have the given tag(s)
G0(Count)	Count of documents returned for the above searches

Table 4.2 Discovery APIs Group 0

Group	Functionality
G1(Relationship)	Search documents that related to the given entity by the specified relation
G1 (Spatial)	Spatial Search using Circle, Line string and Polygon geometry
G1(Text)	Fuzzy search based on given text query
G1(Count)	Count of documents returned for the above searches

Table 4.3 Discovery APIs Group 1

Group	Functionality
Attribute + Geo	Search documents that confine to given geometry and have the specified attribute(s)
Geo + Text	Search documents that confine to given geometry and match a fuzzy text
Attribute + Text	Search documents that have the specified attribute(s) and match a fuzzy text
Attribute + Geo + Text	Search documents that confine to given geometry, have the specified attribute(s) and match a fuzzy text
Tag + Geo	Search documents that confine to given geometry and have the specified tag(s)
Tag + Text	Search documents that have the specified tag(s) and match a fuzzy text
Tag + Geo + Text	Search documents that confine to given geometry, have the specified tag(s) and match a fuzzy text
Response Filtering	Filter the json fields returned in the response (eg. filter value can be [<i>location.geometry</i>] or [<i>deviceModel.modelName</i> , <i>itemCreatedAt</i> , etc.)

Table 4.4 Discovery APIs Group 2

Complex search supports searches using combinations of spatial, text, tag and attribute queries from G0 and G1. In addition to complex search, the G2 supports response filtering which shall also apply for searches defined in G0 and G1.

For each functional profile an abstract suite of test cases has been specified in Section 5. To be conformant to a given functional profile a catalogue service implementation has to pass all the test cases specified for that particular functional profile. Further, a given catalogue service implementation can comply with multiple functional profiles.

It is to be noted that in a complex search, the parameters of complex search are derived using individual parameters from multiple test cases. e.g., a complex search '009a&020a' (Section 5.7.1) represents a single test case. The parameters of this test shall be derived from the parameters of test 009a (Section 5.3.1) and 020a (Section 5.5.3.1). Also note that this is a separate test case and even if an implementation passes 009a and 020a individually it may not necessarily support functionality required by this complex search test case. An example of complex search is given in ANNEXURE-B

5 TEST CASES FOR FUNCTIONAL PROFILES

Every test case verifies the interactions of the implementation of Catalogue Service and the interaction with other DX services.

As mentioned earlier, in addition to the above scenarios, for every successful (2XX) and unsuccessful (4XX) test cases, tests shall be conducted to understand the Data Response and Response URNs which shall be based on the dataset used in the database and the URNs defined in the implementation.

All the functional profile test numbers (001a, 002a.....034d) represented in the tables are associated with a specific test case as defined in the postman collection. The link to the postman collection is mentioned in ANNEXURE-C.

5.1 2XX Test cases for Management APIs

The tests under this section are associated with the Management functional profile as defined in Table 4.1. These tests summarise the HTTP response code 2XX for the create, update, delete and get operations on a catalogue item.

Each operation ensures that the relationship between items are verified. For eg:

- While creating a resource item the corresponding resourceServer, provider and resource group should be present in the Catalogue. This may be performed by verifying the existence of "resourceGroup" and "provider" items in the catalogue system. The

information about the provider and resourceGroup is available in the resource item [See the JSON body of test case 001a in Postman Collection for more information].

- While deleting a provider item the associated resource items and resource groups should be deleted beforehand. This may be performed by verifying the non-existence of “resourceGroup” and “resourceItem(s)” while a provider item is deleted.

Refer to ANNEXURE-A for more information on the relationship between items.

Note: Management API’s can only be accessed by Admins, Providers or Delegates. “resourceItem(s)”, “resourceGroup(s)” and “provider” items can be created, updated or deleted by Providers or Delegates. A “resourceServer(s)” item can be created, updated or deleted only by an Admin.

5.1.1 2XX Test for Management APIs

Functional Profile	Management		
Purpose	To test the correctness of CRUD of a DX document on a Catalogue Service implementation.		
Description	These tests validate the correctness of implementation of upload, update, delete and get operations on a DX document. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
001a	To upload metadata document	<i>Json payload</i>	201
003a	To update existing document		200
005a	To delete existing document	Parameter ‘id’ should be set to have DX entity id	200
007a	To get document of the created DX entity		200

Note: To ease the testing process, the data dump contains a provider, resource group and a resource server item. The resource item available with the postman collection can be used for verifying the Create, Update and Delete APIs links to the above items for verifying the API implementation logic.

5.2 4XX Test cases for Management APIs

The tests under this section are associated with the Management functional profile as defined in Table 4.1. These tests summarise the HTTP response code 4XX for the Create, Update, Delete and Get APIs.

5.2.1 4XX Test for Management APIs

Functional Profile	Management		
Purpose	To test the correctness of CRUD of a DX document on a Catalogue Service implementation for 'invalid parameter(s)' or 'invalid header(s)' or 'invalid json payload' passed for any of the fields in the request.		
Description	These tests validate the correctness of implementation of upload, update, delete and get operations on a DX document. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
002a	This test will return a URN corresponding to invalid schema while uploading a metadata document	<i>Json payload</i>	400
002b	This test will return a URN corresponding to invalid links while uploading a metadata document		400
002c	This test will return a URN corresponding to invalid credentials while uploading a metadata document		401

004a	This test will return a URN corresponding to invalid schema while updating a metadata document	<i>Json payload</i>	400
004b	This test will return a URN corresponding to invalid link while updating a metadata document		400
004c	This test will return a URN corresponding to invalid credentials while updating a metadata document		401
004d	This test will return a URN corresponding to id not found while updating a metadata document		404
006a	This test will return a URN corresponding to invalid credentials while deleting a metadata document	Parameter 'id' should be set to have DX entity id	401
006b	This test will return a URN corresponding to id not being found while deleting a metadata document		404
008a	This test will return a URN corresponding to id not being found while fetching a metadata document	Parameter 'id' should be set to have DX entity id	404

5.3 2XX Test cases for Discovery APIs (G0)

The tests under this section are associated with the G0 (Discovery) functional profile as defined in Table 4.1. These tests summarise the HTTP response code 2XX for the Attribute (Property), List and Tag search implementation.

5.3.1 2XX Test for Attribute (Property) Search

Functional Profile	Discovery APIs (G0)		
Purpose	To test the correctness of Attribute (Property) search for various possibilities of property & value parameters.		
Description	These tests validate the correctness of Attribute (Property) search implementation for various possibilities of property & value parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
009a	This test returns catalogue items that match a property (eg. <i>label</i>) with a given value (eg. <i>environment</i>)	Parameter 'property' shall be set to the key against which search is to be made. Parameter 'value' shall be set to a single value for the key.	200
009b	This test returns catalogue items that match a property (eg. <i>tags</i>) with multiple values (eg. <i>environment</i> or <i>mobility</i>)	Parameter 'property' shall be set to the key against which search is to be made. Parameter 'value' shall be set to multiple values for the key.	200
009c	This test returns catalogue items that match properties (eg. <i>tags</i> , <i>device ids</i>) with multiple values (eg. <i>environment</i> or <i>mobility</i>)	Parameter 'property' shall be set to the keys against which search is to be made. Parameter 'value' shall be set to multiple values for the keys.	200
009d	This test returns catalogue items that match nested properties (eg. <i>location.address</i>) with single (eg. <i>Delhi</i>) or multiple values (eg. <i>Pune</i> or <i>Delhi</i>)	Parameter 'property' shall be set to the keys against which search is to be made. A key can be a sub-property of another property. Parameter 'value' shall be set to multiple values for the keys.	200

5.3.2 2XX Test for List Search

Functional Profile	Discovery APIs (G0)		
Purpose	To test the correctness of List search for various possibilities for the path parameter.		
Description	These tests validate the correctness of List search implementation for various possibilities of the path parameter. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
011a	This test lists all the unique tags across all the catalogue items (eg. AQM, flood, mobility, Swachhata...)	Path parameter is set to 'tags'	200
011b	This test lists all the instances hosted on the catalogue (eg. Pune, Surat, Mumbai..)	Path parameter is set to 'instances'	200
011c	This test lists all the ResourceGroups that are hosted on the catalogue	Path parameter is set to 'resourceGroup'	200
011d	This test lists all the ResourceServers that are hosted on the catalogue	Path parameter is set to 'resourceServer'	200
011e	This test lists all the providers that are registered on the catalogue	Path parameter is set to 'provider'	200

5.3.3 2XX Test for Tag Search

Functional Profile	Discovery APIs (G0)		
Purpose	To test the correctness of Tag (Property) search for various possibilities of property & value parameters.		
Description	These tests validate the correctness of Tag (Property) search implementation for various possibilities of property & value parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
013a	This test will return data based on the property(<i>tag</i>)	Parameter 'property' shall be set to <i>tags</i> . Parameter 'value' shall be set to a single value.	200
013b	This test will return documents where one or more of the provided tags match (eg. <i>[[flood, mobility]]</i>)	Parameter 'property' shall be set to <i>tags</i> . Parameter 'value' shall be set to multiple values. [Documents with at least one of the given tags will be returned]	200

5.3.4 2XX Test for Count Search

Functional Profile	Discovery APIs (G0)		
Purpose	To test the correctness of Count search for various possibilities of property & value parameters.		
Description	These tests validate the correctness of Count search implementation for various possibilities of property & value parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		

Test No.	Operation	Parameters	Response Code
015a	These tests get the total number of hits of documents returned for the corresponding search API [as in sections 5.3.1, 5.3.2 and 5.3.3]	The parameters would be the same as the corresponding search API [as in sections 5.3.1, 5.3.2 and 5.3.3]	200
015b			200
015c			200
015d			200
015e			200
015f			200

5.4 4XX Test cases for discovery (G0)

The tests under this section are associated with the G0(Discovery) functional profile as defined in Table 4.1. These tests summarise the HTTP response code 4XX for the Attribute, List and Tag search implementation.

5.4.1 4XX Test for Attribute (Property) Search

Functional Profile	Discovery APIs (G0)		
Purpose	To test the correctness of Attribute (Property) search for various possibilities of property & value parameters.		
Description	These tests validate the correctness of Attribute (Property) search implementation for various possibilities of property & value parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
010a	This test returns a URN corresponding to invalid id	The parameter 'value' will be set to have a invalid	400

	during a search based on simple attribute	value. Rest of the parameters shall remain same as that for attribute search.	
010b	This test returns a URN corresponding to invalid property value during a search based on simple parameter value	The parameter 'value' will be set to have at least one invalid value. Rest of the parameters shall remain same as that for attribute search.	400
010c	This test returns a URN corresponding to invalid syntax during a search based on simple attribute	The parameter 'property' will be set to have a invalid value. Rest of the parameters shall remain same as that for attribute search.	400
010d	This test returns a URN corresponding to invalid syntax during a search based on multi valued attribute		400
010e	This test returns a URN corresponding to invalid property value during search based on multi attribute	The parameter 'property' will be set to have more than 4 valid values. Rest of the parameters shall remain same as that for attribute search.	400
010f	This test returns a URN corresponding to invalid property value during search based on multi attribute	The parameter 'value' will be set to have more than 4 valid values. Rest of the parameters shall remain same as that for attribute search.	400
010g	This test returns a URN corresponding to invalid property value during search based on multi attribute		400

5.4.2 4XX Test for List Search

Functional Profile	Discovery APIs (G0)		
Purpose	To test the correctness of List search for various possibilities of the path parameter.		
Description	These tests validate the correctness of List search implementation for various possibilities of the path parameter. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameter	Response Code
012a	These tests returns a URN corresponding to invalid syntax during a list search	The path parameter will be set to have mis-spelled value corresponding to the path parameter specified in corresponding list API [as in section 5.3.2]	400
012b			400
012c			400
012d			400
012e			400
012f			400

5.4.4 4XX Test for Tag Search

Functional Profile	Discovery APIs (G0)
Purpose	To test the correctness of Tag (Property) search for various possibilities of property & value parameters.
Description	These tests validate the correctness of Tag (Property) search implementation for various possibilities of property & value parameters. The response will contain results associated with the operation.

Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
014a	These tests return a URN corresponding to invalid syntax during a tag search	The parameter 'value' will be set to have a invalid value. Rest of the parameters shall remain same as that for tag search.	400
014b		The parameter 'property' will be set to have a invalid value. Rest of the parameters shall remain same as that for tag search.	400

5.4.5 4XX Test for Count Search

Functional Profile	Discovery APIs (G0)		
Purpose	To test the correctness of Count search for various possibilities of property & value parameters.		
Description	These tests validate the correctness of Count search implementation for various possibilities of property & value parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
016a	This test returns a URN corresponding to invalid syntax during a count search using tags (single property)	The parameter 'value' will be set to have a invalid value. Rest of the parameters shall remain same as that for tag search.	400
016b	This test returns a URN corresponding to invalid syntax	The parameter 'value' will be set to have a invalid value.	400

	during a count search using tags (single property - multi value)	Rest of the parameters shall remain same as that for tag search.	
016c	This test returns a URN corresponding to invalid syntax during a count search using tags	The parameter 'property' will be set to have a invalid value. Rest of the parameters shall remain same as that for tag search	400
016d	This test returns a URN corresponding to invalid syntax during a count search using attributes (single property)	The parameter 'value' will be set to have a invalid value. Rest of the parameters shall remain same as that for attribute search.	400
016e	This test returns a URN corresponding to invalid syntax during a count search using attributes (single property)	The parameter 'value' will be set to have at least one invalid value. Rest of the parameters shall remain same as that for attribute search.	400
016f	This test returns a URN corresponding to invalid syntax during a count search using attributes	The parameter 'property' will be set to have a invalid value. Rest of the parameters shall remain same as that for attribute search	400

5.5 2XX Test cases for Discovery APIs (G1)

The tests under this section are associated with the G1 (Discovery) functional profile as defined in Table 4.1. These tests summarise the HTTP response code 2XX for Relationship APIs, Relationship Search, Geo-Spatial Search, Text Search and Complex Search implementation.

5.5.1 2XX for Relationship APIs(G1)

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Relationship APIs for various possibilities of id & rel parameters.		
Description	These tests validate the correctness of Relationship APIs' implementation for various possibilities of id & rel parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
017a	These tests return catalogue documents that connected to the input entity by the given relationship	The parameter 'id' will be set to a valid Resource Item id, 'rel' would be set to <i>resourceGroup</i> .	200
017b		The parameter 'id' will be set to a valid Resource Group id, 'rel' would be set to <i>resource</i> .	200
017c		The parameter 'id' will be set to a valid Resource Item id, 'rel' would be set to <i>provider</i> .	200
017d		The parameter 'id' will be set to a valid Resource Group id, 'rel' would be set to <i>provider</i>	200
017e		The parameter 'id' will be set to a valid Resource Item id, 'rel' would be set to <i>resourceServer</i>	200
017f		The parameter 'id' will be set to a valid Resource Group id, 'rel'	200

		would be set to <i>resourceServer</i> .	
017g	Returns the type of an catalogue item (data descriptor)	The parameter 'id' will be set to a valid Resource Item id, 'rel' would be set to <i>type</i> .	200
017h		The parameter 'id' will be set to a valid Resource Group id, 'rel' would be set to <i>type</i> .	200

5.5.2 2XX for Relationship Search(G1)

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Relationship search for various possibilities of relationship & value parameters.		
Description	These tests validate the correctness of Relationship search implementation for various possibilities of relationship & value parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
019a	These tests return catalogue documents that satisfy a relationship value provided in the query	The parameter 'relationship' will be set to have attribute values against which query is to be made. The parameter 'value' will be set to have a value to which the attribute has to be matched. Both the parameters can have multiple values.	200
019b			200

5.5.3 2XX for Geo-spatial Search (G1)

5.5.3.1 2XX for Point (Circle) Search

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Geospatial (Point) search for various possibilities of geospatial parameters.		
Description	These tests validate the correctness of Geospatial (Point) search implementation for various possibilities of geoproperty, georel, maxDistance, geometry & coordinates parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
020a	This test returns catalogue documents that fall within a given circular(point and radius) geometry	The parameter 'geoproperty' will be set to <i>location</i> , 'georel' will be set to <i>within</i> , 'maxDistance' will be set to a number between 0 and 10000 (in meters), 'geometry' will be set to <i>Point</i> , and coordinates will be set as [<i>lon, lat</i>]	200

5.5.3.2 2XX for Polygon Search

Functional Profile	Discovery APIs (G1)
Purpose	To test the correctness of Geospatial (Polygon) search for various possibilities of geospatial parameters.
Description	These tests validate the correctness of Geospatial (Polygon) search implementation for various possibilities of geoproperty, georel, geometry & coordinates parameters. The response will contain results associated with the operation.

Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
022a	This test returns catalogue documents that fall within/intersects/disjoint from the input polygon perimeter	The parameter 'geoproperty' will be set to <i>location</i> , 'georel' can be set to <i>within</i> or <i>intersects</i> or <i>disjoint</i> , 'geometry' will be set to <i>Polygon</i> , and coordinates will be set as <i>[[lon1, lat1],[lon2,lat2]...[lon1,lat1]]</i> . [Note that the first and last (lon,lat) will be the same to complete the polygon.]	200

5.5.3.3 2XX for Line String Search

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Geospatial (LineString) search for various possibilities of geospatial parameters.		
Description	These tests validate the correctness of Geospatial (LineString) search implementation for various possibilities of geoproperty, georel, geometry & coordinates parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
024a	This test returns catalogue documents that fall on the input linestring	The parameter 'geoproperty' will be set to <i>location</i> , 'georel' will be set to <i>intersects</i> , 'geometry' will be set to <i>LineString</i> , and	200

		coordinates will be set as [[lon1, lat1],[lon2,lat2]...[lon10,lat10]].	
--	--	--	--

5.5.3.4 2XX for BBox Search

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Geospatial (BBox) search for various possibilities of geospatial parameters.		
Description	These tests validate the correctness of Geospatial (BBox) search implementation for various possibilities of geoproperty, georel, geometry & coordinates parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
026a	This test returns catalogue documents that fall within the input geojson BBox	The parameter 'geoproperty' will be set to <i>location</i> , 'georel' will be set to <i>within</i> , 'geometry' will be set to <i>BBox</i> , and coordinates will be set as [[lon1,lat1],[lon2,lat2]] where first pair of coordinates	200

5.5.4 2XX for Text Search (G1)

Functional Profile	Discovery APIs (G1)
Purpose	To test the correctness of Text search for various possibilities of the parameter q.
Description	These tests validate the correctness of Text search implementation for various possibilities of the parameter q. The response will contain results associated with the operation.

Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
028a	These tests return catalogue documents based on the fuzzy text based query	The parameter 'q' will be set to a fuzzy text phrase	200
028b		The parameter 'q' will be set to include special characters	200

5.5.5 2XX for Count APIs (G1)

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Count search for various possibilities of parameters corresponding to the respective search API.		
Description	These tests validate the correctness of Count search implementation for various possibilities of parameters corresponding to the respective search API. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
030a	These tests get the total number of hits of documents for the corresponding search API [as in sections 5.5.1-5.5.4]	The parameters would be the same as the corresponding search API [as in sections 5.5.1-5.5.4]	200
030b			200
030c			200
030d			200
030e			200
030f			200

5.6. 4XX Test cases for Discovery (G1)

The tests under this section are associated with the G1 (Discovery) functional profile as defined in Table 4.1. These tests summarise the HTTP response code 4XX for Relationship APIs, Relationship Search, Geo-Spatial Search, Text Search and Complex Search implementation.

5.6.1 4XX Tests for Relationship APIs (G1)

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Relationship APIs for various possibilities of id & rel parameters.		
Description	These tests validate the correctness of Relationship APIs' implementation for various possibilities of id & rel parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
018a	This test returns a URN corresponding to invalid param value during a relationship API call	The parameter 'rel' will be set to have a mis-spelled value. Rest of the parameters shall remain same as that for Relationship search.	400
018b			400
018c	This test returns a URN corresponding to missing params during a relationship API call	The parameter 'rel' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Relationship API.	400
018d		The parameter 'id' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Relationship API.	400

5.6.2 4XX Tests for Geo-Spatial Search(G1)

5.6.2.1 4XX Tests for Point (Circle) Search(G1)

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Geospatial(Point) search for various possibilities of geospatial parameters.		
Description	These tests validate the correctness of Geospatial(Point) search implementation for various possibilities of geoproperty, georel, maxDistance, geometry & coordinates parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
021a	This test returns a URN corresponding to invalid geo value during a geo-spatial search (georel value is incorrect)	The parameter 'georel' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (Point) search.	400
021b	This test returns a URN corresponding to invalid syntax during a geo-spatial search (coordinates value is incorrect)	The parameter 'coordinate' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (Point) search.	400
021c	This test returns a URN corresponding to invalid geo value during a geo-spatial search (geoproperty value is incorrect)	The parameter 'geoproperty' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (Point) search.	400
021d	This test returns a URN corresponding to invalid syntax during a geo-spatial search(geoproperty is mis-spelt)	The parameter 'geoproperty' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (Point) Search.	400
021e	This test returns a URN corresponding to invalid syntax	The parameter 'georel' will not be set (The test requires a mandatory	400

	during a geo-spatial search(georel is mis-spelt)	parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (Point) Search.	
021f	This test returns a URN corresponding to invalid syntax during a geo-spatial search(geometry is mis-spelt)	The parameter 'geometry' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (Point) Search..	400
021g	This test returns a URN corresponding to invalid property value during a geo-spatial search(max decimal precision can be upto 6 digits)	The parameter 'coordinates' is set to have floating point value with precision more than 6. Rest of the parameters shall remain same as that for Geospatial(Point) Search.	400
021h	This test returns a URN corresponding to invalid property value during a geo-spatial search(max number of coordinates for a geo search is 10)	The parameter 'coordinates' is set to have more than 10 valid coordinate pairs. Rest of the parameters shall remain same as that for Geospatial(Point) Search.	400
021i	This test returns a URN corresponding to invalid property value during a geo-spatial search(max distance for a geo search is 10000m)	The parameter 'maxDistance' is set to have a value more than 10000 m. Rest of the parameters shall remain same as that for Geospatial(Point) Search.	400
021j	This test returns a URN corresponding to invalid property value during a geo-spatial search(distance for geo-spatial search cannot be negative)	The parameter 'maxDistance' is set to have a negative value. Rest of the parameters shall remain same as that for Geospatial(Point) Search.	400
021k	This test returns a URN corresponding to invalid property value during a geo-spatial search(distance for geo-spatial search cannot be infinity)	The parameter 'maxDistance' is set to have a very large number as the value. Rest of the parameters shall remain same as that for Geospatial(Point) Search.	400

5.6.2.2 4XX Tests for Polygon Search(G1)

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Geospatial (Polygon) search for various possibilities of geospatial parameters.		
Description	These tests validate the correctness of Geospatial (Polygon) search implementation for various possibilities of geoproperty, georel, geometry & coordinates parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
023a	This test returns a URN corresponding to invalid geo value during a geo-spatial search	The parameter 'geoproperty' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (Polygon) search.	400
023b	This test returns a URN corresponding to invalid geo value during a geo-spatial search	The parameter 'georel' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (Polygon) search.	400
023c	This test returns a URN corresponding to invalid syntax during a geo-spatial search	The parameter 'coordinate' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (Polygon) search.	400
023d	This test returns a URN corresponding to invalid syntax during a geo-spatial search(geoproperty is misspelt)	The parameter 'geoproperty' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (Polygon) Search..	400

023e	This test returns a URN corresponding to invalid syntax during a geo-spatial search(georel is mis-spelt)	The parameter 'georel' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (Polygon) Search.	400
023f	This test returns a URN corresponding to invalid syntax during a geo-spatial search(coordinates is mis-spelt)	The parameter 'coordinates will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (Polygon) Search..	400
023g	This test returns a URN corresponding to invalid property value during a geo-spatial search(max decimal precision can be upto 6 digits)	The parameter 'coordinates' is set to have floating point value with precision more than 6. Rest of the parameters shall remain same as that for Geospatial(Polygon) Search.	400
023h	This test returns a URN corresponding to invalid property value during a geo-spatial search(max number of coordinates for a geo search is 10)	The parameter 'coordinates' is set to have more than 10 valid coordinate pairs. Rest of the parameters shall remain same as that for Geospatial(Polygon) Search.	400
023i	This test returns a URN corresponding to invalid property value during a geo-spatial search(distance for geo-spatial search cannot be infinity)	The parameter 'coordinate' is set to have a very large number as one of the values. Rest of the parameters shall remain same as that for Geospatial(Polygon) Search.	400

5.6.2.3 4XX Tests for Linestring Search(G1)

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Geospatial (LineString) search for various possibilities of geospatial parameters.		
Description	These tests validate the correctness of Geospatial (LineString) search implementation for various possibilities of geoproperty, georel, geometry & coordinates parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
025a	This test returns a URN corresponding to invalid geo value during a geospatial search	The parameter 'geoproperty' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (LineString) search.	400
025b	This test returns a URN corresponding to invalid geo value during a geospatial search	The parameter 'georel' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (LineString) search.	400
025c	This test returns a URN corresponding to invalid syntax during a geospatial search	The parameter 'coordinate' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (LineString) search.	400
025d	This test returns a URN corresponding to invalid syntax during a geospatial search(geoproperty is mis-spelt)	The parameter 'geoproperty' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (LineString) Search..	400
025e	This test returns a URN corresponding to invalid syntax during a geo-	The parameter 'georel' will not be set (The test requires a mandatory parameter to be missing). Rest of the	400

	spatial search(georel is mis-spelt)	parameters shall remain same as that for Geospatial (LineString) Search.	
025f	This test returns a URN corresponding to invalid syntax during a geo-spatial search(coordinates is mis-spelt)	The parameter 'coordinates will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (LineString) Search..	400
025g	This test returns a URN corresponding to invalid property value during a geo-spatial search(max decimal precision can be upto 6 digits)	The parameter 'coordinates' is set to have floating point value with precision more than 6. Rest of the parameters shall remain same as that for Geospatial(LineString) Search.	400
025h	This test returns a URN corresponding to invalid property value during a geo-spatial search(max number of coordinates for a geo search is 10)	The parameter 'coordinates' is set to have more than 10 valid coordinate pairs. Rest of the parameters shall remain same as that for Geospatial(LineString) Search.	400
025i	This test returns a URN corresponding to invalid property value during a geo-spatial search(distance for geo-spatial search cannot be infinity)	The parameter 'coordinate' is set to have a very large number as one of the values. Rest of the parameters shall remain same as that for Geospatial(LineString) Search.	400
Methodology			
Execute an API call with parameters that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes			

5.6.2.1 4XX Tests for Bbox Search(G1)

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Geospatial (BBox) search for various possibilities of geospatial parameters.		
Description	These tests validate the correctness of Geospatial (BBox) search implementation for various possibilities of geoproperty, georel, geometry & coordinates parameters. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
027a	This test returns a URN corresponding to invalid geo value during a geo-spatial search	The parameter 'geoproperty' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (BBox) search.	400
027b	This test returns a URN corresponding to invalid geo value during a geo-spatial search	The parameter 'georel' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (BBox) search.	400
027c	This test returns a URN corresponding to invalid syntax during a geo-spatial search	The parameter 'coordinate' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (BBox) search.	400
027d	This test returns a URN corresponding to invalid syntax during a geo-spatial search(geoproperty is misspelt)	The parameter 'geoproperty' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (BBox) Search..	400

027e	This test returns a URN corresponding to invalid syntax during a geo-spatial search(georel is mis-spelt)	The parameter 'georel' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (BBox) Search.	400
027f	This test returns a URN corresponding to invalid syntax during a geo-spatial search(coordinates is mis-spelt)	The parameter 'coordinates' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (BBox) Search..	400
027g	This test returns a URN corresponding to invalid property value during a geo-spatial search(max decimal precision can be upto 6 digits)	The parameter 'coordinates' is set to have floating point value with precision more than 6. Rest of the parameters shall remain same as that for Geospatial(BBox) Search.	400
027h	This test returns a URN corresponding to invalid property value during a geo-spatial search(max number of coordinates for a geo search is 10)	The parameter 'coordinates' is set to have more than 10 valid coordinate pairs. Rest of the parameters shall remain same as that for Geospatial(BBox) Search.	400
027i	This test returns a URN corresponding to invalid property value during a geo-spatial search(distance for geo-spatial search cannot be infinity)	The parameter 'coordinate' is set to have a very large number as one of the values. Rest of the parameters shall remain same as that for Geospatial(BBox) Search.	400

5.6.3 4XX Tests for Text Search(G1)

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Text search for various possibilities of the parameter q.		
Description	These tests validate the correctness of Text search implementation for various possibilities of the parameter q. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
029a	This test returns a URN corresponding to invalid syntax during a text search using special characters (illegal characters in query)	The parameter 'q' is set to have invalid characters in its string value.	400
029b	This test returns a URN corresponding to invalid syntax during a text search (mandatory field is not provided)	The parameter 'q' will not be set (The test requires a mandatory parameter to be missing).	400
029c	This test returns a URN corresponding to invalid property value(query string exceeds maximum size)	The parameter 'q' will be set to have more than 100 valid characters in its string value.	400

5.6.4 4XX Tests for Count APIs(G1)

Functional Profile	Discovery APIs (G1)		
Purpose	To test the correctness of Count search for various possibilities of parameters corresponding to the respective search API.		
Description	These tests validate the correctness of Count search implementation for various possibilities of parameters corresponding to the respective search API. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
031a	This test returns a URN corresponding to invalid syntax (illegal characters in query)	The parameter 'q' is set to have invalid characters in its string value.	400
031b	This test returns a URN corresponding to invalid syntax (mandatory field is not provided)	The parameter 'q' will not be set (The test requires a mandatory parameter to be missing).	400
031c	This test returns a URN corresponding to invalid geo value (geoproperty/georel/geometry value is incorrect)	The parameter 'georel' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (Point) search.	400
031d	This test returns a URN corresponding to invalid syntax search(geoproperty/georel/geometry is mis-spelt)	The parameter 'geoproperty' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (Point) Search.	400
031e	This test returns a URN corresponding to invalid geo value (geoproperty/georel/geometry value is incorrect)	The parameter 'geoproperty' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (Polygon) search.	400

031f	This test returns a URN corresponding to invalid syntax search(geoproperty/georel/geometry is mis-spelt)	The parameter 'geoproperty' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (Polygon) Search.	400
031g	This test returns a URN corresponding to invalid geo value (geoproperty/georel/geometry value is incorrect)	The parameter 'geoproperty' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (LineString) search.	400
031h	This test returns a URN corresponding to invalid syntax search(geoproperty/georel/geometry is mis-spelt)	The parameter 'geoproperty' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (LineString) Search.	400
031i	This test returns a URN corresponding to invalid geo value (geoproperty/georel/geometry value is incorrect)	The parameter 'geoproperty' is set to have a invalid value. Rest of the parameters shall remain same as that for Geospatial (BBox) search.	400
031j	This test returns a URN corresponding to invalid syntax search(geoproperty/georel/geometry is mis-spelt)	The parameter 'geoproperty' will not be set (The test requires a mandatory parameter to be missing). Rest of the parameters shall remain same as that for Geospatial (BBox) Search.	400

5.7. 2XX Test cases for Discovery APIs(G2)

The tests under this section are associated with the G2 (Discovery) functional profile as defined in Table 4.1. These tests summarise the HTTP response code 2XX for Complex Search implementation.

5.7.1 2XX for Complex Search

Functional Profile	Discovery APIs (G2)		
Purpose	To test the correctness of Complex search for various possibilities of parameters corresponding to the respective search APIs.		
Description	These tests validate the correctness of Complex search implementation for various possibilities of parameters corresponding to the respective search APIs. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Parameters	The parameters will be set according to the tests that are part of the given complex search combination.		
Test No.	Operation	Combination of tests	Response Code
032a	Complex search can be a combination query of any of the above search (discovery) profiles	(009a or 009b or 009c or 009d) and (020a or 022a or 024a or 026a)	200
032b		(009a or 009b or 009c or 009d) and (028a or 028b)	200
032c		(009a or 009b or 009c or 009d) and (020a or 022a or 024a or 026a) and (028a or 028b)	200
032d		(013a or 013b) and (020a or 022a or 024a or 026a)	200
032e		(013a or 013b) and (028a or 028b)	200

032f		(013a or 013b) and (020a or 022a or 024a or 026a) and (028a or 028b)	200
032g		(020a or 022a or 024a or 026a) and (028a or 028b)	200

5.7.2 2XX Test cases for Response Filtering

The following tests show the implementation of Response Filtering (RF) available for the corresponding tests in G0,G1 and G2.

These tests also include limit + offset on the set of documents returned. Limit here is defined as the threshold on the number of documents returned and offset is defined as the number of documents to skip from the first document from the set of documents returned.

It is to be noted that the list of tests below is not exhaustive.

Functional Profile	Discovery APIs (G2)		
Purpose	To test the correctness of Response Filtering for various search APIs.		
Description	These tests validate the correctness of Response Filtering implementation for various possibilities of parameters corresponding to the respective search APIs. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
033a	These tests return selected fields of items, where the items have given property	The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033b		The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents.	200

		The parameters 'limit' and 'offset' will be set to numeric values. Rest of the parameters shall remain the same as that of the corresponding search API.	
033c	This test returns list of given type, but returns documents based on the limit specified	The parameter 'limit' will be set to a numeric value. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033d	These tests return selected fields of items, where the items match given tag(s)	The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033e		The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. The parameters 'limit' and 'offset' will be set to numeric values. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033f	These tests return selected fields of items, where the items satisfy given relationship	The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033g		The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. The parameters 'limit' and 'offset' will be set to numeric values. Rest of the parameters shall remain the same as that of the corresponding search API.	200

033h	This test returns document that satisfy given relation but returns documents based on the limit + offset specified	The parameters 'limit' and 'offset' will be set to numeric values. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033i	These tests return selected fields of items, where the items confine to given geometry (Point)	The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033j		The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. The parameters 'limit' and 'offset' will be set to numeric values. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033k	These tests return selected fields of items, where the items confine to given geometry (Polygon)	The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033l		The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. The parameters 'limit' and 'offset' will be set to numeric values. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033m	These tests return selected fields of items, where the items confine to given geometry	The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033n		The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents.	200

	(LineString)	The parameters 'limit' and 'offset' will be set to numeric values. Rest of the parameters shall remain the same as that of the corresponding search API.	
033o	These tests return selected fields of items, where the items confine to given geometry (BBox)	The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033p		The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. The parameters 'limit' and 'offset' will be set to numeric values. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033q	These tests return selected fields of items, where the items satisfy given fuzzy text search	The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033r		The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. The parameters 'limit' and 'offset' will be set to numeric values. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033s	These tests return selected fields of items, where the items satisfy the combination of search parameters	The parameter 'filter' is set to have a list of values that are a subset of the fields of the returned documents. Rest of the parameters shall remain the same as that of the corresponding search API.	200
033t			200
033u			200
033v			200

5.8. 4XX Test Cases for Discovery(G2)

The tests under this section are associated with the G2(Discovery) functional profile as defined in Table 4.1. These tests summarise the HTTP response code 4XX for Response Filtering implementation.

5.8.1 4XX Tests for Response filtering (G2)

Functional Profile	Discovery APIs (G2)		
Purpose	To test the correctness of Response Filtering for various search APIs.		
Description	These tests validate the correctness of Response Filtering implementation for various possibilities of parameters corresponding to the respective search APIs. The response will contain results associated with the operation.		
Methodology	Execute an API call that shall return a successful and verifiable response. Verify that output data is as per the test design. Optionally, one can also verify that the response contains appropriate URN codes		
Test No.	Operation	Parameters	Response Code
034a	This test returns a URN corresponding to bad filtering during response filtering (max number of filters can be 10)	The parameter 'filter' is set to have a list of more than 10 values. Rest of the parameters shall remain the same as that of the corresponding search API.	400
034b	This test returns a URN corresponding to invalid syntax during response filtering	The parameter 'filter' is set to have a list of values with at least one invalid value. Rest of the parameters shall remain the same as that of the corresponding search API.	
034c	This test returns a URN corresponding to invalid property value during response filtering	The parameter 'limit' is set to have a large value such that limit+offset>10000. Rest of the parameters shall remain the same as that of the corresponding search API.	400
034d	This test returns a URN corresponding to invalid	The parameter 'offset' is set to have a large value such that	400

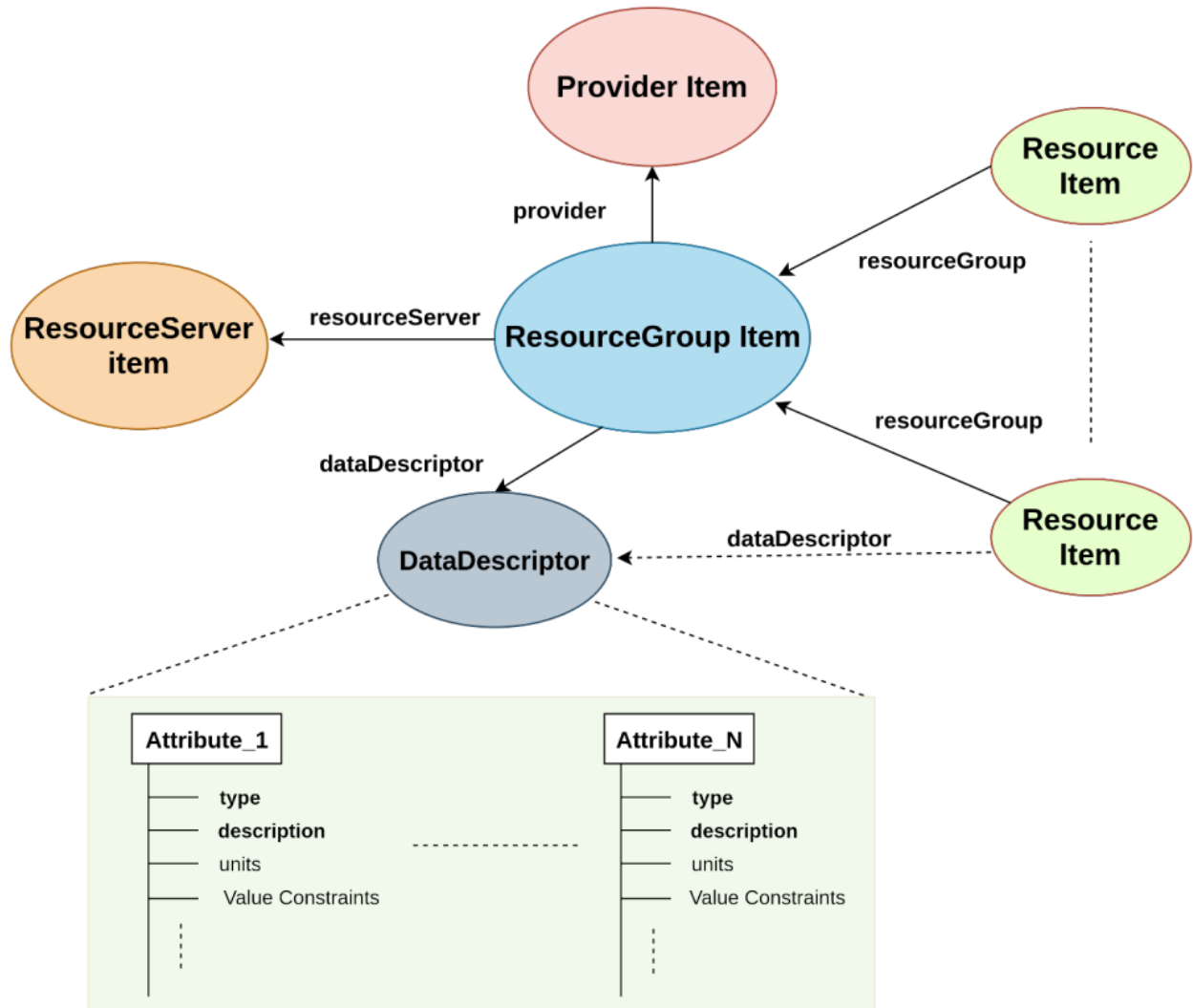
	property value during response filtering	limit+offset>10000. Rest of the parameters shall remain the same as that of the corresponding search API.	
--	---	--	--

DRAFT

ANNEXURES

ANNEX-A

A DX Catalogue Service defines a base set of item types namely Resource Server (RS), Provider, Resource Group (RG) and Resource Item (RI); as mentioned in section 5.1.2 of IS 18003 (Part 2) specification¹. The relationship between each of the items is mapped in Fig.2 of IS 18003 (Part 2). For ease of understanding, it is recaptured below



This annexure presents the requirements necessary to maintain the relationship between all the item types.

- ❖ The items can only be created in this order
 - Resource Server
 - Provider
 - Resource Group
 - Resource Item
- ❖ A Resource Server can only be created by an admin
- ❖ A Resource server can be considered as a parent to Provider, a Provider can be considered as a parent to a Resource Group and so on.
- ❖ While updating an item the parent id cannot be changed
- ❖ The items can only be deleted in this order
 - Resource Item
 - Resource Group
 - Provider
 - Resource Server
- ❖ A Resource Server can only be deleted by an admin

Note: A postman sub-collection can be found for this annexure attached to the postman collection containing test cases for management and discovery APIs.

ANNEX-B

This annexure presents an example to conduct a complex test based on the combination of tests in Group 0 and Group 1. Note that this is a separate test case and even if an implementation passes the tests individually it may not necessarily support functionality required by this complex search test case.

A complex test using G0(Attribute:MultiValue) and G1 (GeoSpatial:Polygon) and G1(Text:Text) will have the parameters for an attribute search on a 'polygon' spatial search along with a text search. This test is a combination of three test 009b as in Section 5.3.1 & 022a as in section 5.5.3 & test 028a as in section 5.5.4. The parameters for this complex test are shown below

An example of a Complex test.	
Parameter	Parameter Value
property	"[location.address]"
value	"[[pune,delhi]]"
geoproperty	location
georel	within
geometry	Polygon
coordinates	[[[73.696,18.592],[73.69079,18.391],[73.96,18.3643],[74.09,18.526],[73.8947,18.689830],[73.696,18.592]]]
q	"Sens data"

ANNEX-C

The postman collection consisting of test cases are available [here](#)

ANNEX-D

The data dump for the postman collection for executing the tests are available [here](#)