

# OGC API - Tiles - 3D (GeoVolumes) Engineering Report

Publication Date: YYYY-MM-DD

Approval Date: YYYY-MM-DD

Submission Date: 2020-07-15

Reference number of this document: OGC 20-030

Reference URL for this document: <http://www.opengis.net/doc/PER/20-030>

Category: OGC Public Engineering Report

Editors: Timothy Miller and Gil Trenum

Title: OGC API - Tiles - 3D (GeoVolumes) Engineering Report

---

## **OGC Public Engineering Report**

### **COPYRIGHT**

Copyright © 2020 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

### **WARNING**

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Public Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

## LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for

complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

# Table of Contents

<b>1. Subject</b>	<b>6</b>
<b>2. Executive Summary</b>	<b>7</b>
2.1. Document contributor contact points	7
2.2. Foreword	8
<b>3. References</b>	<b>9</b>
<b>4. Terms and definitions</b>	<b>10</b>
4.1. Abbreviated terms	10
<b>5. Overview</b>	<b>12</b>
<b>6. Background</b>	<b>13</b>
6.1. Existing Standards/Previous Work	13
<b>7. Draft specification</b>	<b>15</b>
<b>8. Scope</b>	<b>21</b>
<b>9. Conformance</b>	<b>22</b>
<b>10. References</b>	<b>23</b>
<b>11. Terms and definitions</b>	<b>24</b>
<b>12. Conventions</b>	<b>25</b>
12.1. Identifiers	25
12.2. Abbreviated terms	25
<b>13. Introduction</b>	<b>27</b>
13.1. Overview	27
13.2. Use cases	27
13.2.1. A 3D client	27
<b>14. GeoVolumes API</b>	<b>33</b>
14.1. Requirements Class "Core"	33
14.1.1. API landing page	35
14.1.2. Declaration of conformance classes	36
14.1.3. API definition	36
14.1.4. Collections	37
14.1.5. 3D-Container	39
14.2. Requirements Class "Extension"	42
14.2.1. Requirements Class "spatial query extension"	43
<b>15. Media Types</b>	<b>46</b>
15.1. application/json+i3s	46
15.2. application/json+3dtiles	46
<b>Annex A: Conformance Class Abstract Test Suite (Normative)</b>	<b>48</b>
A.1. Conformance Class "Core"	48
A.1.1. Test Case 1	48
A.1.2. Test Case 2	48
<b>Annex B: Revision History</b>	<b>50</b>



# Chapter 1. Subject

This Engineering Report documents the draft specification for a three-dimensional (3D) geodata Application Programming Interface (API) that organizes access to a variety of 2D / 3D datasets and their distributions according to a nested hierarchy of 3D geospatial volumes (GeoVolumes). The GeoVolumes (initially Tiles-3D / 3D Container) API specification is consistent with OGC API - Common and supports both link-follow and bbox query methods of access to resources of interest.

[1] Source: **Tiles-3D Community Standard** [<https://www.ogc.org/standards/3DTiles>] 3D Tiles Community Standard 1.0 (18-053r2)

[2] Source: **Tiles-3D Community Standard** [<https://www.ogc.org/standards/3DTiles>] 3D Tiles Community Standard 1.0 (18-053r2)

# Chapter 2. Executive Summary

The goal of the 3D Data Container and Tiles API Pilot was to explore an integrated suite of draft specifications for 3D geospatial resources compatible with existing OGC 3D delivery standards in order to support smooth transitions between 2D and 3D environments; allow applications working with 2D tile resources to get 3D tiled resources; and enable 3D bounding volumes to support multiple data containers, datasets, and distributions. To achieve these goals, the Pilot developed a draft GeoVolumes API providing browse and query access to 3D geospatial data for streamed data delivery by means of nested geospatial volume (originally 3D container) resources termed GeoVolumes. The 3D data resources supported by this API include feature geometries, feature attribute values, elevation models, texture data, etc.. The API provides both link-follow and Bounding Box (bbox) query methods of access to 2D and 3D resources in a manner independent of the underlying data store, supporting multiple standard geospatial distribution formats such as **3D Tiles** [<https://www.ogc.org/standards/3DTiles>], **I3S** [<https://www.ogc.org/standards/i3s>], **CDB** [<https://www.ogc.org/standards/cdb>], and **CityGML** [<https://www.ogc.org/standards/citygml>].

A number of data server and client implementations were developed in the course of the Pilot in order to test interoperable data delivery via the draft GeoVolumes API. The draft API developed by the Pilot has been defined using the OpenAPI 3.0 definition language and conforms to the **OGC API - Common** [<https://ogcapi.ogc.org/common>] building blocks such as landing page, API definition, conformance declaration, and collections. The Pilot developed and tested the GeoVolumes API in order to advance open-standard and unified approaches for delivering 3D content using state of the art API practices that work across different data formats, streaming protocols, and model types.

The following requirements were identified in the Call for Participation (CFP) of the Pilot. The API shall:

1. Provide API access to tiled and un-tiled 3D resources
2. Allow exchange of content organized according to nested 3D Container (now GeoVolume) resource types developed in this Pilot
3. Align with existing and emerging OGC APIs, standards, and candidate standards (e.g. OGC API - Features and OGC API - Common, 3D Tiles Community Standard, I3S Community Standard)

While the present draft specification developed for the GeoVolumes API includes all functionality implemented and tested during the Pilot, further work is needed to extend the API for additional data types and nested volume schemes. Future work should also address the integration of the GeoVolumes API with other OGC API building blocks to provide seamless interaction from navigating volumes of interest to interaction with specific dataset formats and distributions.

## 2.1. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:



## Contacts

Name	Organization	Role
Ryan Gauthier	US Army Geospatial Center	Sponsor/Contributor
Jeff Harrison	US Army Geospatial Center	Sponsor/Contributor
Tom Myers	US Army Geospatial Center	Sponsor/Contributor
Tim Miller	Leidos	Editor
Gil Trenum	Leidos	Editor
Josh Lieberman	OGC	Editor
Terry Idol	OGC	Contributor
Rob Jones	Helyx	Contributor
Matthew Knight	Helyx	Contributor
Anneley Hadland	Helyx	Contributor
Tamrat Belayneh	ESRI	Contributor
Jérôme Jacovella-St-Louis	Ecere	Contributor
Aaron Brinton	Cognitics	Contributor
Michala Hill	Cognitics	Contributor
Ignacio Correas	Skymantics	Contributor
Preston Rodrigues	Steinbeis	Contributor
Kevin Ring	Cesium	Contributor
Volker Coors	Steinbeis	Contributor
Thunyathep Santhanavanich	Steinbeis	Contributor

## 2.2. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

# Chapter 3. References

The following normative documents are referenced in this document.

- OGC: OGC 18-053r2, 3D Tiles Community Standard 1.0 [<https://www.ogc.org/standards/3DTiles>]
- OGC: OGC 17-014r5, OGC Indexed 3d Scene Layer (I3S) and Scene Layer Package Format Specification [<http://www.ogc.org/standards/i3s>]
- OGC: OGC 17-069r3, OGC API - Features - Part 1: Core [<http://docs.openeospatial.org/is/17-069r3/17-069r3.html>]
- IETF: RFC 7946 - The GeoJSON Format [<https://tools.ietf.org/html/rfc7946>]

# Chapter 4. Terms and definitions

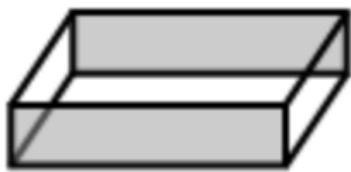
For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact\_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

- *coordinate reference system*

coordinate system that is related to the real world by a datum term name (source: ISO 19111)

- *Bounding Volume*

Typically a simple shape like a sphere, rectangular box, or convex hull that can simply be tested for intersection or overlap. <sup>[3]</sup>



box



region



sphere

- *YAML*

YAML is a human friendly data serialization standard for all programming languages.

## 4.1. Abbreviated terms

- API Application Programming Interface
- CDB Common Database
- COTS Commercial Off The Shelf
- CRS Coordinate Reference Systems
- B3DM Batched 3D Model
- BIM Building Information Model
- BVH Bounding Volume Hierarchy
- DCE Distributed Computing Environment
- glTF GL Transmission Format
- I3DM Instanced 3D Model
- JSON JavaScript Object Notation

- MBV Minimum Bounding Volume
- MBS Minimum Bounding Sphere
- LOD Level of Detail
- OBB Oriented Bounding Box
- RS Regular Subdivision
- TMS Tile Map Services
- YAML A recursive acronym for "YAML Ain't Markup Language"
- 3DC 3D-Container
- 3DGV 3D GeoVolume

[3] Source: **Tiles-3D Community Standard** [<https://www.ogc.org/standards/3DTiles>] 3D Tiles Community Standard 1.0 (18-053r2)

# Chapter 5. Overview

Section 6 provides the background of the Pilot effort and cites existing relevant standards and existing work.

Section 7 begins the draft specification for the GeoVolumes API.

Section 8 provides a high-level description of scope of the GeoVolumes API.

Section 9 defines the requirements/conformance classes for the GeoVolumes API.

Section 10 cites relevant references for the ER.

Section 11 provides relevant terms and definitions.

Section 12 provides relevant identifiers and abbreviated terms.

Section 13 describes client/server interactions via the API Sequence Diagram.

Section 14 provides an overview of resources and applicable HTTP methods for the GeoVolumes API.

Section 15 describes the JSON media types used for returning specific content and the HTML media type for web pages provided by the API.

Annex A describes conformance class test cases.

# Chapter 6. Background

The 3D Data Container and Tiles API Pilot builds on previous work from other OGC Innovation Program initiatives as well as OGC Standards Program efforts. The following documents serve as foundational references for this Pilot activity. Some items listed below are not yet released to the public. Draft versions of these documents have been made available. The final versions of these documents may change from the currently provided versions. Individuals are advised to check the OGC website for updates on these documents.

## 6.1. Existing Standards/Previous Work

- 3D Tiles Community Standard 1.0 (18-053r2) - An OGC standard for streaming and rendering massive 3D geospatial content such as Photogrammetry, 3D Buildings, BIM/CAD, Instanced Features, and Point Clouds. The standard defines a hierarchical data structure and a set of tile formats which deliver content that can be rendered. Of specific interest to this initiative is glTF. The standard document also describes 3D Tile Styles, a declarative styling specification which may be applied to tilesets.
- OGC Indexed 3D Scene Layer (I3S) and Scene Layer Package Format Specification (OGC 17-014r5) - An OGC Community standard which provides the delivery format and persistence model for Scene Layers. A Scene Layer is a container for arbitrarily large amounts of heterogeneously distributed 3D geographic content, supporting coordinate reference systems and height models in conjunction with a rich set of layer types.
- OGC Testbed-13: 3D Tiles and I3S Interoperability and Performance Engineering Report (OGC 17-046) - The report captures the lessons learned from prototyping interoperability of 3D Tiles and I3S using: a 3D Portrayal Service, performance studies of 3D tiling algorithms, and a proof-of-concept of the use of 3D Tiles and I3S as data delivery formats for the OGC 3D Portrayal Service interface standard [1].
- OGC 3D Portrayal Service 1.0 Standard (OGC 15-001r4) - The OGC standard specifies how geospatial 3D content is described, selected, and delivered. The standard provides a framework to determine whether 3D content is interoperable at the content representation level.
- OpenAPI (v3.0) - OpenAPI is a freely-available API description framework that provides a developer with programmatic access to a software application or service. These APIs use sets of technologies that enable websites and/or client applications to interact with each other by using REST, SOAP, JavaScript, and other web technologies. These APIs have allowed web communities to create an open architecture for sharing content and data between communities and applications. A very typical application for an open API is to access data: Tweets, geolocation(s), maps, stock quotes, weather sensors, and so forth. OGC members and staff have actively been investigating OpenAPI (and its commercial equivalent, Swagger) since 2016. It was recognized that the existing web service standards were in effect web APIs, but that modernizing their means of providing content to the web required a fundamental change in underlying design. Two documents advanced the idea further: 1) the OGC Open Geospatial APIs White Paper and the Spatial Data on the Web Best Practices, the latter jointly developed by OGC and W3C. These documents highlighted how geospatial data

should be more native to the web. Further, OGC staff were working on “implementer-friendly” views of our standards and experimented with an OpenAPI definition for the Web Map Tile Service (WMTS) and became aware of work by the United Kingdom Hydrographic Office (UKHO) to publish OpenAPI definitions for OGC Web Map Service (WMS) of Electronic Navigational Charts.

- The OGC API - Features - Part 1: Core standard (OGC 17-069r3): This standard provides building blocks to enable APIs to create, modify and query features on the Web. This standard specifies the fundamental API building blocks for interacting with features. The spatial data community uses the term 'feature' for things in the real world that are of interest. The standard is part of the OGC API family of standards. OGC API standards define modular API building blocks to spatially enable Web APIs in a consistent way. Most recently, the OGC Web Feature Service (WFS) and Filter Encoding Service (FES) Standards Working Group rebuilt the WFS standard to the new OGC API - Features standard with an integrated OpenAPI definition as core to describe how to build against the standard. The results are being taken by others working to advance other OGC standards as OpenAPI definitions. This work can be reviewed on the [OGC API website](https://ogcapi.ogc.org) [https://ogcapi.ogc.org].
- OGC Testbed-15: Maps and Tiles API Engineering Report (OGC 19-069) - This Engineering Report was developed as part of the Testbed-15 initiative [2]. It is not a standard, but reflects the latest work on Web APIs to access and manage maps and 2D tiled data.
- OGC Testbed-15: Styles API Engineering Report (OGC 19-010r2) - This Engineering Report was developed as part of the Testbed-15 initiative. It is not a standard, but reflects the latest work done in OGC on styles for 2D data. The Styles API is a Web API that enables map servers and clients as well as visual style editors to manage and fetch styles [3]. The API is consistent with the emerging OGC API family of standards. The API complements the Features, Maps and Tiles APIs and builds on the conceptual model for styles developed in OGC Testbed-15. This report specifies the API using OpenAPI, specifies how the same styles should be represented in a GeoPackage and documents the lessons learned during the implementation.
- OGC® Routing Pilot ER (OGC 19-041r3) - This Engineering Report was produced by the OGC Routing Pilot. The Pilot assessed an abstract baseline suite of functions, capabilities, and encodings to address a common standard interface for network routing functionality [4]. This included guidance for extending the Routing API to account for various routing data models and support for network routing engine configuration via the Routing API. The engineering report is not an OGC standard, but it was delivered to the OGC Standards Program for further consideration.

# Chapter 7. Draft specification



## Open Geospatial Consortium

Submission Date: yyyy-mm-dd

Approval Date: yyyy-mm-dd

Publication Date: yyyy-mm-dd

External identifier of this OGC® document: <http://www.opengis.net/doc/IS/ogcapi-geovolumes/1.0>

Internal reference number of this OGC® document: 20-030

Version: **1.0.0-SNAPSHOT (Editor's draft)** [<http://docs.opengeospatial.org/DRAFTS/20-030.html>]

Category: OGC® Implementation Standard

Editor: Tim Miller

### OGC API - GeoVolumes, Part 1: Core

#### Copyright notice

Copyright © 2020 Open Geospatial Consortium

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>

#### Warning

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OGC® Implementation Standard

Document subtype: Interface

Document stage: Draft

Document language: English

## License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

## i. Abstract

This document is a draft specification of the GeoVolumes Application Programming Interface (API) that defines a Web API enabling servers and 3D clients to access multiple 3D model formats that represent features and models common to specific 3D geospatial volumes of interest. The Web API promotes interoperability providing the capabilities for 3D clients to access multiple servers and 3D formats (e.g. I3S, 3D Tiles, and CityGML). It is consistent with the emerging OGC API family of standards.

The GeoVolumes API supports several types of consumers, mainly:

- geospatial server that provides 3D model content;
- map clients that render 3D model content;
- the modeling and simulation community;
- the gaming community.

It implements the conceptual API and model as documented in chapter 7 of the "3D Data Container Engineering Report". Note that the "3D Data Container (3dcontainer)" is a precursor term for "GeoVolume (geovolume)" used in this and companion reports. "GeoVolume" stands for "Geospatial Volume".

The model defines the following key concepts:

- GeoVolumes (3DGV) are the main resources.
- 3DGV information documents are further extensions of the collection information documents defined in the present draft OGC API Common Collections extension.
- 3DGV are each characterized by defined 3D extents.
- 3DGV can be organized in a flat list or as a bounding volume hierarchy.

This model directly maps to the resources and documents in the GeoVolumes API, which supports the resources and operations listed in the [Table 1](#) below.

Resource	Path	HTTP method	Document reference
Landing Page	/	GET	<a href="#">API landing page</a>
Conformance Declaration	<a href="#">/conformance</a>	GET	<a href="#">Declaration of conformance classes</a>
API Definition	<a href="#">/api</a>	GET	<a href="#">API definition</a>
Collections	<a href="#">/collections</a>	GET	<a href="#">Collections</a>

Resource	Path	HTTP method	Document reference
3D Container	<code>/collections/{containerID}</code>	GET	3D-Container

Table 1. GeoVolumes API - overview of resources and applicable HTTP methods

In order to support 3DGV, data APIs (for example, supporting OGC API Features and/or the draft OGC API Tiles) require additional capabilities, too. These are:

- Access applicable 3DGV/3D model content (path `/collections/{containerID}`).
- Query 3DGV/3D model content based on a bounding box (path `/collections?bbox=x1,y1,z1,x2,y2,z2``, `/collections/{containerID}?bbox=x1,y1,z1,x2,y2,z2`).

This document uses **OpenAPI 3.0** [<http://spec.openapis.org/oas/v3.0.2>] to specify the building blocks of the API.

The OpenAPI Specification (OAS) defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection. When properly defined, a consumer can understand and interact with the remote service with a minimal amount of implementation logic.

An OpenAPI definition can then be used by documentation generation tools to display the API, code generation tools to generate servers and clients in various programming languages, testing tools, and many other use cases.

— OpenAPI Specification, Introduction

**ii. Keywords** The following are keywords to be used by search engines and document catalogues.

OGC document, GeoVolumes API, Geospatial Volume API, API, OpenAPI, JSON, HTML, I3S, 3D Tiles, CityGML, glTF

### iii. Preface

Multiple 3D geospatial client applications and 3D model formats exist; OGC is currently missing a robust conceptual model and API to facilitate access and interoperability between these applications and 3D model formats. This document specifies draft building blocks for Web APIs to

fetch 3D GeoVolume content.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the draft specification set forth in this document, and to provide supporting documentation.

#### **iv. Submitting organizations**

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- U.S. Army Geospatial Center (AGC)

#### **v. Submitters**

All questions regarding this submission should be directed to the editor or the submitters:

<b>Name</b>	<b>Affiliation</b>
Tim Miller ( <i>editor</i> )	Leidos
Ryan Gauthier	U.S. Army Geospatial Center (AGC)
Jeff Harrison	U.S. Army Geospatial Center (AGC)

# Chapter 8. Scope

The GeoVolumes Application Programming Interface (API) is a Web API that enables servers and clients to access 3D content in various formats.

The GeoVolumes API (aka OGC API - GeoVolumes) is part of the emerging OGC API family of standards. The API complements the current and emerging OGC API specifications for features, maps and tiles and builds on the conceptual model for the encoding of styles and their metadata developed in OGC Testbed-20.

The building blocks of the API are specified using OpenAPI 3.0.

# Chapter 9. Conformance

This draft specification defines the following requirements/conformance classes for the GeoVolumes API:

- "core" provides access to 3D content. JSON is a mandatory encoding in requests and responses where JSON schemas have been specified for the GeoVolumes API.

There are four requirements/conformance classes that extend the information about Collection resources as specified in OGC API - Features - Part 1: Core:

- "collectionType" adds information on the collection type (default: "3dcontainer").
- "contentExtent" adds information on the minimum 3D spatial extent of dataset content, using either a box, region, or sphere.
- "children" adds an array of zero or more "child" 3dcontainers.
- "content" adds an array of zero or more dataset distribution content references.

The standardization target for all classes is: Web API.

Conformance with this draft specification shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site.

In order to conform minimally to this draft specification, a software implementation has to implement the "core" requirements class.

# Chapter 10. References

The following normative documents are referenced in this document.

- **OGC: OGC 18-053r2, 3D Tiles Community Standard 1.0** [<https://www.ogc.org/standards/3DTiles>]
- **OGC: OGC 17-014r5, OGC Indexed 3d Scene Layer (I3S) and Scene Layer Package Format Specification** [<http://www.ogc.org/standards/i3s>]
- **IETF: RFC 7946 - The GeoJSON Format** [<https://tools.ietf.org/html/rfc7946>]



# Chapter 11. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact\_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

- *coordinate reference system*

coordinate system that is related to the real world by a datum term name (source: ISO 19111)

- *Bounding Volume*

Typically a simple shape like a sphere, rectangular box, or convex hull that can simply be tested for intersection or overlap. <sup>[4]</sup>



box



region



sphere

- *YAML*

YAML is a human friendly data serialization standard for all programming languages.

[4] Source: [Tiles-3D Community Standard](https://www.ogc.org/standards/3DTiles) [https://www.ogc.org/standards/3DTiles] 3D Tiles Community Standard 1.0 (18-053r2)

# Chapter 12. Conventions

This section provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

## 12.1. Identifiers

The normative provisions in this draft specification are denoted by the URI:

<http://www.opengis.net/spec/ogcapi-geovolumes-1/1.0>

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

## 12.2. Abbreviated terms

*NOTE: The abbreviated terms clause gives a list of the abbreviated terms and the symbols necessary for understanding this document. All symbols should be listed in alphabetical order. Some more frequently used abbreviated terms are provided below as examples.*

- API Application Programming Interface
- CDB Common Database
- COTS Commercial Off The Shelf
- CRS Coordinate Reference Systems
- B3DM Batched 3D Model
- BIM Building Information Model
- BVH Bounding Volume Hierarchy
- DCE Distributed Computing Environment
- glTF GL Transmission Format
- I3DM Instanced 3D Model
- JSON JavaScript Object Notation
- MBV Minimum Bounding Volume
- MBS Minimum Bounding Sphere
- LOD Level of Detail
- OBB Oriented Bounding Box
- RS Regular Subdivision

- TMS Tile Map Service

# Chapter 13. Introduction

## 13.1. Overview

This document specifies draft building blocks for the GeoVolumes API to fetch GeoVolume content. The API provides access to 2D and 3D resources and allows the exchange of content compliant with the GeoVolume developed in this Pilot. The API aligns with existing and emerging OGC API standards. Note that this documentation still uses the term "3D Container", that has been superseded by and is equivalent to "GeoVolume".

## 13.2. Use cases

This section describes expectations of how clients and servers will interact with the GeoVolumes API.

The following use cases assume that:

- The URI of the landing page of the GeoVolumes API is <http://example.org/>

### 13.2.1. A 3D client

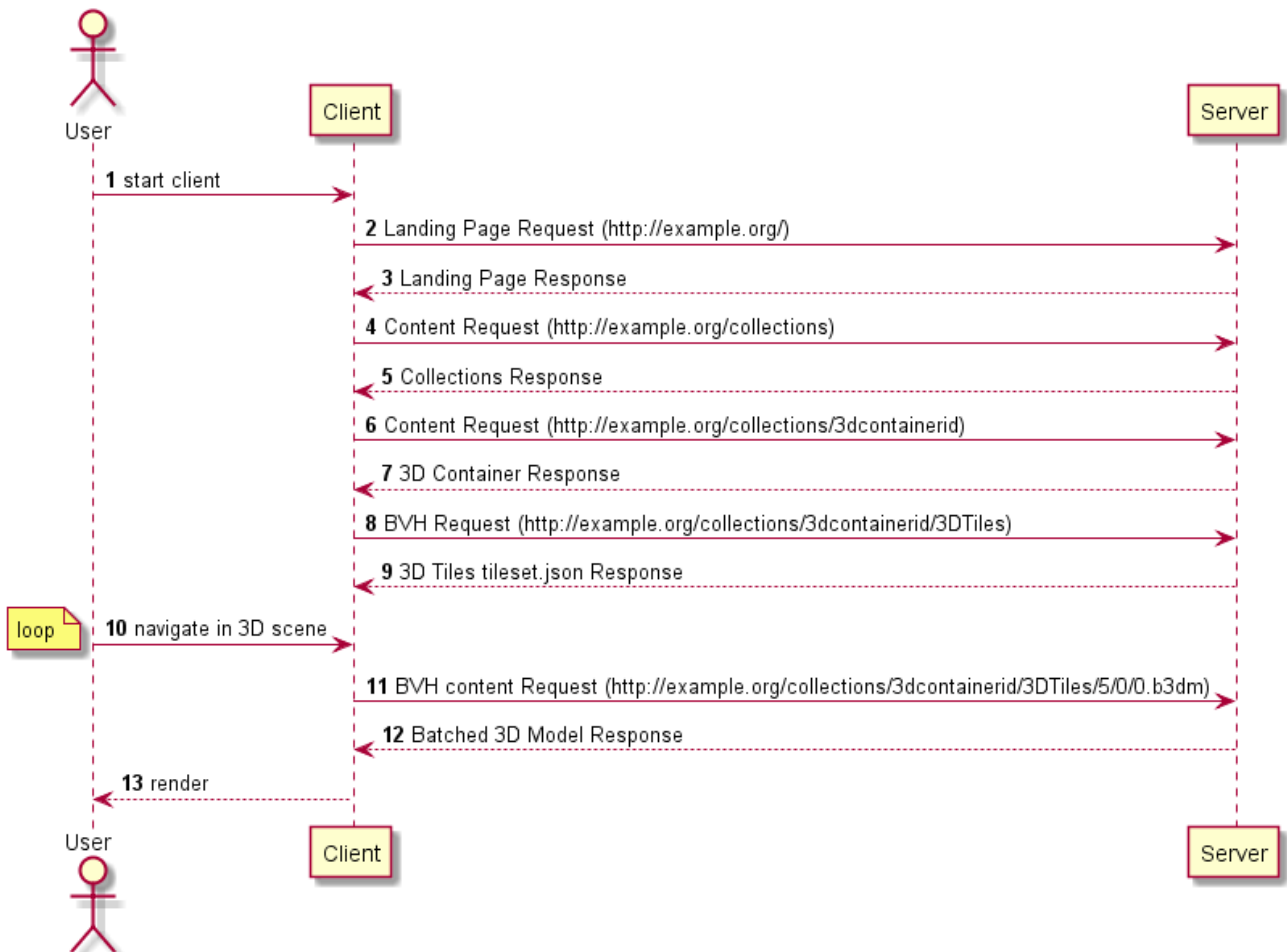


Figure 1. API Sequence Diagram

The following is the user/client/server sequence for accessing a 3D Container:

1. User starts client, selects provider, and loads collections.
2. Client Landing Page Request. Client requests the landing page from server. Client landing page request: <http://example.org/>
3. Server Landing Page Response. The server provides the following landing page response:

```
{
  "title": "Pilot 3D Container API",
  "description": "A pilot of an API for 3D containers and tiles.",
  "links": [
    {service-desc},
    {conformance},
    {
      "type": "application/json",
      "title": "Collections",
      "href": "http://example.org/collections/",
      "rel": "data"
    }
  ]
}
```

4. Client Content Request. Client parses the landing page for data link relationship and requests the collections from the server. The client can optionally provide a bounding box query string to filter the contents from the server and format type. Client collections request: <http://example.org/collections>.
5. Sever Collections Response. The server processes the request returning a collections object, containing an array of links and array of 3d containers. The response may contain multiple containers/container ids in a flat or hierarchical organization.
6. Client Content Request. The client parses the collections response, within the 3d container. Links to the 3D Tiles content maybe located in (1) links, (2) content, or (3) children properties. From the 3D container information, the client can process either a flat list or hierarchical list of 3D containers. The organization depends on the use of the children object within the 3D container. Client container request: <http://example.org/NewYork/NewYork-buildings/>
7. Server 3D Container Response. The server responds with only one 3D Container from the container ID ("NewYork/NewYork-buildings"). The container can contain multiple 3D containers, with multiple formats, and with a flat or hierarchical organization.

```

{
  "id": "NewYork/NewYork-buildings",
  "title": "NYC - 3D Buildings Manhattan",
  "description": "3D Buildings in Manhattan, New York.",
  "collectionType": "3d-container",
  "extent": {
    "spatial": [
      -74.01900887327089,
      40.700475291581974,
      -11.892070104139751,
      -73.9068954348699,
      40.880256294183646,
      547.7591871983744
    ],
    "crs": "http://www.opengis.net/def/crs/OGC/0/CRS84h"
  },
  "links": [
    {
      "rel": "self",
      "href": "http://example.org/collections/NewYork/NewYork-buildings/",
      "type": "application/json",
      "title": "NYC - 3D Buildings Manhattan"
    },
    {
      "rel": "items",
      "href": "http://example.org/collections/NewYork/NewYork-buildings/i3s/"
    },
    {
      "type": "application/json+i3s",
      "title": "NYC - 3D Buildings Manhattan: i3s"
    },
    {
      "rel": "items",
      "href": "http://example.org/collections/NewYork/NewYork-buildings/3dTiles/",
      "type": "application/json+3dtiles",
      "title": "NYC - 3D Buildings Manhattan: 3D Tiles"
    }
  ],
  "content": [
    {
      "title": "NYC - 3D Buildings Manhattan: i3s",
      "rel": "original",
      "href": "http://example.org/collections/NewYork/NewYork-buildings/i3s/"
    },
    {
      "type": "application/json+i3s",
      "collectionType": "3d-container"
    }
  ],
}

```

```
{
  "title": "NYC - 3D Buildings Manhattan: 3D Tiles",
  "rel": "original",
  "href": "http://example.org/NewYork/NewYork-buildings/3dTiles/",
  "type": "application/json+3dtiles",
  "collectionType": "3d-container"
}
]
}
```

8. Client Bounding Volume Hierarchy (BVH) Request. The client parses the 3D container for 3D Tiles link. Client 3D Tiles/tileset.json request : <http://example.org/collections/NewYork/NewYork-buildings/3dTiles/>
9. Server 3D tileset.json Response. The server response with tileset.json BVH.

```

{
  "asset": {
    "version": "1.0",
    "extras": {
      "ion": {
        "georeferenced": true,
        "movable": false
      }
    }
  },
  "properties": {
    "Height": {
      "maximum": 547.7591871983744,
      "minimum": -11.892070104139751
    },
    "Latitude": {
      "maximum": 40.880256294183646,
      "minimum": 40.700475291581974
    },
    "Longitude": {
      "maximum": -73.9068954348699,
      "minimum": -74.01900887327089
    }
  },
  "geometricError": 740.0197559011849,
  "root": {
    "boundingVolume": {
      "region": [
        -1.29187544264487,
        0.7103573144863446,
        -1.289919109210917,
        0.7134950819190251,
        0,
        547.6909683533274
      ]
    },
    "geometricError": 740.0197559011849,
    "refine": "ADD",
    "children": []
  }
}

```

10. User navigates through the 3D scene

11. Client BVH Content Request. The client requests 3D Tiles batched 3D model from the server.  
 Client B3DM request: <http://example.org/collections/NewYork/NewYork-buildings/3dTiles/>



5/0/0.b3dm

12. Server Batched 3D Model Response. Server provides batched 3D model content to the client.
13. Client renders 3D model content to the user.

# Chapter 14. GeoVolumes API

## NOTE

This clause specifies the GeoVolumes API as designed and implemented in the 3D Data Container & Tiles API Pilot task of OGC Testbed 20.

The GeoVolumes API, an OpenAPI 3.0 REST API, conforms to the OGC API-Common foundation resources: landing page, API definition, conformance, and collections (spatial resources). It provides access to 3D resources and allows the exchange of content compliant with the 3D Data Container (3DC).

The GeoVolumes API supports the resources and operations listed in the Table below with the associated conformance class and the link to the document section that specifies the requirements.

Resource	Path	HTTP method	Document reference
Landing Page	/	GET	<a href="#">API landing page</a>
Conformance Declaration	<a href="#">/conformance</a>	GET	<a href="#">Declaration of conformance classes</a>
API Definition	<a href="#">/api</a>	GET	<a href="#">API definition</a>
Collections	<a href="#">/collections</a>	GET	<a href="#">Collections</a>
3D Container	<a href="#">/collections/{3DContainerID}</a>	GET	<a href="#">3D-Container</a>

Table 2. GeoVolumes API - overview of resources and applicable HTTP methods

This version of the GeoVolumes API was written with the following assumptions:

- The data provider may choose to implement flat list or hierarchical list of 3D containers.

## 14.1. Requirements Class "Core"

## NOTE

An "OGC API - Common" specification is under development (October 1, 2019). The current draft of Common is based on the generic concepts of the [OGC API - Features - Part 1: Core](#) standard which are presently being developed in the OGC API - Common specification. However, the work is in the earliest stages of the standardization process. In order to avoid duplicating content, this document does not copy the basic requirements, recommendations and permissions from OGC API Common/Features. If "OGC API - Common" is not available as a normative reference, the alternative would be to remove the dependency and to specify the following normative statements are part of this requirements class:

- Landing page
  - Requirement [/req/core/root-op](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req_core_root-op) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req\_core\_root-op]
  - Requirement [/req/core/root-success](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req_core_root-success) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req\_core\_root-success]
    - Change: No **data** link to **/collections** is required, but a **collections** link has to point to the **/collections** resource
- API definition
  - Requirement [/req/core/api-definition-op](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req_core_api-definition-op) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req\_core\_api-definition-op]
  - Permission [/per/core/api-definition-uri](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#per_core_api-definition-uri) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#per\_core\_api-definition-uri]
  - Requirement [/req/core/api-definition-success](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req_core_api-definition-success) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req\_core\_api-definition-success]
  - Recommendation [/rec/core/api-definition-oas](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#rec_core_api-definition-oas) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#rec\_core\_api-definition-oas]
- Conformance declaration
  - Requirement [/req/core/conformance-op](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req_core_conformance-op) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req\_core\_conformance-op]
  - Requirement [/req/core/conformance-success](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req_core_conformance-success) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req\_core\_conformance-success]
- Web API
  - Requirement [/req/core/http](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req_core_http) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req\_core\_http]
  - Recommendation [/rec/core/head](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#rec_core_head) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#rec\_core\_head]
  - Permission [/per/core/additional-status-codes](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#per_core_additional-status-codes) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#per\_core\_additional-status-codes]
  - Requirement [/req/core/query-param-unknown](http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req_core_query-param-unknown) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req\_core\_query-param-unknown]

- **Requirement** [/req/core/query-param-invalid](#) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#req\_core\_query-param-invalid]
- **Recommendation** [/rec/core/etag](#) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#rec\_core\_etag]
- **Recommendation** [/rec/core/cross-origin](#) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#rec\_core\_cross-origin]
- **Recommendation** [/rec/core/link-header](#) [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#rec\_core\_link-header]

### 14.1.1. API landing page

The landing page provides links to the API definition, the Conformance statements and the metadata about the 3D Container in this dataset.

*Example 1. Landing page in JSON*

```
{
  "tile": "OGC 3D Container/API Pilot",
  "description": "A Pilot of anAPI for 3D Containers",
  "links": [
    {
      "href": "http://data.example.org/",
      "rel": "service-desc",
      "type": "application/json",
      "title": "Service Description"
    },
    {
      "href": "http://data.example.org/conformance",
      "rel": "conformance",
      "type": "application/json",
      "title": "Conformance"
    },
    {
      "href": "http://data.example.org/collections",
      "rel": "data",
      "type": "application/json",
      "title": "Collections"
    }
  ]
}
```

### 14.1.2. Declaration of conformance classes

The following is an example of the conformance declaration of a 3D Container API that implements all requirements classes except "html".

*Example 2. Conformance declaration in JSON*

```
{
  "conformsTo": [
    "http://www.opengis.net/spec/ogcapi-geovolumes-1/1.0/conf/core",
    "http://www.opengis.net/spec/ogcapi-geovolumes-1/1.0/conf/oas30",
    "http://www.opengis.net/spec/ogcapi-geovolumes-1/1.0/conf/geojson"
  ]
}
```

### 14.1.3. API definition

The following is an example of the API definition.

```

{
  "links": [
    {
      "href": "http://data.example.org/",
      "rel": "self",
      "type": "application/json",
      "title": "this document"
    },
    {
      "href": "http://data.example.org/api",
      "rel": "service",
      "type": "application/openapi+json;version=3.0",
      "title": "the API definition"
    },
    {
      "href": "http://data.example.org/conformance",
      "rel": "conformance",
      "type": "application/json",
      "title": "conformance classes implemented by this server"
    },
    {
      "href": "http://data.example.org/collections",
      "rel": "data",
      "type": "application/json",
      "title": "Metadata about the collections"
    }
  ]
}

```

#### 14.1.4. Collections

Describes the collection of 3D Containers. Optional query items: bounding box and return format type.

Requirement 1	/req/core/collections-op
A	The server SHALL support the HTTP GET operation at the path <a href="#">/collections</a> .

<b>Requirement 2</b>	<b>/req/core/collections-success</b>
A	A successful execution of the operation SHALL be reported as a response with an HTTP status code <b>200</b> .
B	The content of that response SHALL conform to the media type stated in the Content-Type header.
C	The content of that response SHALL conform to the media type stated in the query string.
D	The content of that response SHALL be constrained by the bbox stated in the query string.

E	<p>The content of that response SHALL be based upon the following OpenAPI 3.0 schema:</p> <pre> type: object properties:   links:     type: array     items:       type: object       required:         - href         - rel       properties:         href:           type: string         title:           type: string           nullable: true         rel:           type: string         type:           type: string           nullable: true         hreflang:           type: string           nullable: true       collections:         type: array         items:           \$ref: '3dcontainer' </pre>
---	--

### 14.1.5. 3D-Container

<b>Requirement 3</b>	<b><code>/req/core/collections/{containerid}-op</code></b>
A	The server SHALL support the HTTP GET operation at the path <code>/collections/{containerid}</code> .
<b>Requirement 4</b>	<b><code>/req/core/collections/{containerid}-success</code></b>



A	A successful execution of the operation SHALL be reported as a response with an HTTP status code <b>200</b> .
B	The content of that response SHALL conform to the media type stated in the Content-Type header.
C	The content of that response SHALL conform to the media type stated in the query string.
D	The content of that response SHALL be constrained by the bbox stated in the query string.

E

The content of that response SHALL be based upon the following OpenAPI 3.0 schema:

```
type: object
required:
  - id
  - extent
  - links
properties:
  id:
    type: string
  title:
    type: string
    nullable: true
  description:
    type: string
    nullable: true
  collectionType:
    type: string
    default: '3d-container'
  itemType:
    type: string
    default: 'unknown'
  extent:
    type: object
    properties:
      spatial:
        type: object
        properties:
          bbox:
            type: array
            minItems: 4
            maxItems: 6
            items:
              type: number
          crs:
            type: string
            default:
              'http://www.opengis.net/def/crs/OGC/1.3/CRS84'
          temporal:
            type: object
            properties:
              interval:
                type: array
                nullable: true
                minItems: 1
                items:
                  type: array
                  minItems: 2
```

## 14.2. Requirements Class "Extension"

This clause specifies the extensions to the `Collection` [http://docs.opengeospatial.org/is/17-069r3/17-069r3.html#core-overview] as additional requirements/conformance classes to OGC API - GeoVolumes.

The extensions are the following:

- The collections (path `/collections`) is extended by the addition of a bounding box query parameter.
- The collection (path `/collections/{containerId}`) is extended by the addition of a bounding box query parameter.

This resulting API has the resources listed in the Table below.

Resource	Path	HTTP method	Changes
Landing Page	<code>/</code>	GET	unchanged
API Definition	<code>/api</code>	GET	API definition
Conformance Declaration	<code>/conformance</code>	GET	unchanged, except for returning additional conformance classes
Collections	<code>/collections?bbox</code>	GET	Added bounding box parameter
3D Container	<code>/collections/{containerId}?bbox</code>	GET	Added bounding box parameter

Table 3. Overview of resources, applicable HTTP methods

The following is an example of the conformance declaration of a GeoVolumes API that implements all requirements classes.

```

- href
- rel
properties:
  href:
    type: string
  title:
    type: string
    nullable: true
  rel:
    type: string
  type:
    type: string
    nullable: true
  hreflang:
    type: string
    nullable: true
children:
  type: array
  items:
    $ref: 3dcontainer

```

```
{
  "conformsTo": [
    "http://www.opengis.net/spec/ogcapi-geovolumes-1/1.0/conf/core",
    "http://www.opengis.net/spec/ogcapi-geovolumes-1/1.0/conf/oas30",
    "http://www.opengis.net/spec/ogcapi-geovolumes-1/1.0/conf/html",
    "http://www.opengis.net/spec/ogcapi-geovolumes-1/1.0/conf/spatialquery"
  ]
}
```

type: string

### 14.2.1. Requirements Class "spatialquery extension"

type: string

#### Requirements Class

<http://www.opengis.net/t15/geovolume/1.0/req/spatialquery>

Target type Web API

Dependency

Requirement	Path
<b>Requirement 5</b>	<b>/req/spatialquery/op</b>
A	The server SHALL support the HTTP GET operation at the path <a href="#">/collection?bbox</a> for a 3D Container.
<b>Requirement 6</b>	<b>/req/spatialquery/op</b>
A	The server SHALL support the HTTP GET operation at the path <a href="#">/collection/{containerId}?bbox</a> for each 3D Container.
<b>Requirement 7</b>	<b>/req/spatialquery/success</b>
A	A successful execution of the operation SHALL be reported as a response with a HTTP status code <b>200</b> .

B

The content of that response SHALL be based upon the following OpenAPI 3.0 schema:

```
type: object
required:
  - id
  - extent
  - links
properties:
  id:
    type: string
  title:
    type: string
    nullable: true
  description:
    type: string
    nullable: true
  collectionType:
    type: string
    default: '3d-container'
  itemType:
    type: string
    default: 'unknown'
  extent:
    type: object
    properties:
      spatial:
        type: object
        properties:
          bbox:
            type: array
            minItems: 4
            maxItems: 6
            items:
              type: number
          crs:
            type: string
            default:
              'http://www.opengis.net/def/crs/OGC/1.3/CRS84'
          temporal:
            type: object
            properties:
              interval:
                type: array
                nullable: true
                minItems: 1
                items:
                  type: array
                  minItems: 2
```

C	The <code>id</code> member of each spatial query SHALL be unique.
---	---

```

type: string
format: date-

time

nullable:

true

trs:
  type: string
  nullable: true
  default:
'http://www.opengis.net/def/uom/ISO-8601/0/Gregorian'
contentExtent:
  type: array
  nullable: true
  items:
    type: number
    format: double
    minItems: 4
    maxItems: 12
crs:
  type: string
  default:
'http://www.opengis.net/def/crs/OGC/1.3/CRS84'
links:
  type: array
  items:
    type: object
    required:
      - href
      - rel
    properties:
      href:
        type: string
      title:
        type: string
        nullable: true
      rel:
        type: string
      type:
        type: string
        nullable: true
      hreflang:
        type: string
        nullable: true
    children:
      type: array
      items:
        $ref: 3dcontainer

```

# Chapter 15. Media Types

The following media types are applicable to the GeoVolumes API:

- `application/json` is the JSON media type used for all content.
- `text/html` is the HTML media type for all "web pages" provided by the API.

## 15.1. `application/json+i3s`

- Type name: application
- Subtype name: json+i3s
- Required parameters: n/a
- Optional parameters: n/a
- Encoding considerations: See RFC 8259, The JavaScript Object Notation (JSON) Data Interchange Format
- Security considerations: See Section 12 of RFC 8259
- Interoperability considerations: n/a
- Published specification: link:
- Applications that use this media type:
- Additional information:
  - Deprecated alias names for this type: n/a
  - Magic number(s): n/a
  - File extension(s): .json
  - Macintosh file type code(s): n/a
- Person to contact for further information: n/a
- Intended usage: COMMON
- Restrictions on usage: none
- Author: n/a
- Change controller:

## 15.2. `application/json+3dtiles`

- Type name: application

- Subtype name: json+3dtiles
- Required parameters: n/a
- Optional parameters: n/a
- Encoding considerations: See RFC 8259, The JavaScript Object Notation (JSON) Data Interchange Format
- Security considerations: See Section 12 of RFC 8259
- Interoperability considerations: n/a
- Published specification: link:
- Applications that use this media type:
- Additional information:
  - Deprecated alias names for this type: n/a
  - Magic number(s): n/a
  - File extension(s): .json
  - Macintosh file type code(s): n/a
- Person to contact for further information: n/a
- Intended usage: COMMON
- Restrictions on usage: none
- Author: n/a
- Change controller:

```

content:
  type: array
  items:
    type: object
    required:
      - href
      - rel
properties:
  href:
    type: string
  title:
    type: string
  nullable: true
  rel:
    type: string
  type:
    type: string
    nullable: true
  hreflang:
    type: string
    nullable: true

```



# Annex A: Conformance Class Abstract Test Suite (Normative)

## A.1. Conformance Class "Core"

### A.1.1. Test Case 1

<b>Test id:</b>	
<b>Requirement(s):</b>	/req/core/collections-op, /req/core/collections-success
<b>Test purpose:</b>	Verify that the collections resources can be fetched.
<b>Test method:</b>	<ol style="list-style-type: none"><li>1. Issue an HTTP GET request to the path <code>/collections</code> with header <code>Accept: application/json</code>.</li><li>2. Issue an HTTP GET request to the path <code>/collections?f=json</code>.</li><li>3. Issue an HTTP GET request to the path <code>/collections?bbox=x1,y1,{z1},x2,y2,{z2}</code>.</li><li>4. Validate that the response has a status code 200.</li><li>5. Validate the contents of the returned document against the schema in: <code>/collections</code>, item E.</li><li>6. Verify that each container id <code>#/collections/{i}</code> (where <code>{i}</code> is the index of the style in the array) is unique.</li><li>7. Verify that each collections has at least one link with <code>rel=self</code>.</li><li>8. Verify that for each link with <code>rel=self</code> that the <code>href</code> value links to a resource at the path <code>/collections/{containerid}</code> where <code>{containerid}</code> is the <code>id</code> member of the 3d container.</li></ol>

### A.1.2. Test Case 2

<b>Test id:</b>	
<b>Requirement(s):</b>	/req/core/collections/{containerid}-op, /req/core/collections/{containerid}-success
<b>Test purpose:</b>	Verify that the 3d container resources can be fetched.

<b>Test method:</b>	<ol style="list-style-type: none"><li>1. Issue an HTTP GET request to the path <code>/collections/{containerid}</code> with header <code>Accept: application/json</code>.</li><li>2. Issue an HTTP GET request to the path <code>/collections/{containerid}?f=json</code>.</li><li>3. Issue an HTTP GET request to the path <code>/collections/{containerid}?bbox=x1,y1,{z1},x2,y2,{z2}</code>.</li><li>4. Validate that the response has a status code 200.</li><li>5. Validate the contents of the returned document against the schema in: <code>/collections/{containerid}</code>, item E.</li><li>6. Verify that each container id <code>#/collections/{i}</code> (where <code>{i}</code> is the index of the style in the array) is unique.</li><li>7. Verify that each collection has at least one link with <code>rel=self</code>.</li><li>8. Verify that for each link with <code>rel=self</code> that the <code>href</code> value links to a resource at the path <code>/collections/{containerid}</code> where <code>{containerid}</code> is the <code>id</code> member of the 3d container.</li></ol>
---------------------	---

# Annex B: Revision History

Date	Editor	Release	Primary clauses modified	Description
2020-07-15	T. Miller	0.1	All	Submitted to TC
2020-10-06	G. Hobona	0.2	Multiple	Final staff review

# Annex C: Bibliography

[1] Coors, V.: OGC Testbed-13: 3D Tiles and I3S Interoperability and Performance ER. OGC 17-046, Open Geospatial Consortium, <https://docs.ogc.org/per/17-046.html> (2018).

[2] Maso Pau, J.: OGC Testbed-15: Maps and Tiles API Engineering Report. OGC 19-069, Open Geospatial Consortium, <http://docs.opengeospatial.org/per/19-069.html> (2019).

[3] Portele, C.: OGC Testbed-15: Styles API Engineering Report. OGC 19-010r2, Open Geospatial Consortium, <http://docs.opengeospatial.org/per/19-010r2.html> (2019).

[4] Meek, S., Brown, T., Portele, C.: OGC® Routing Pilot ER. OGC 19-041r3, Open Geospatial Consortium, <http://docs.opengeospatial.org/per/19-041r3.html> (2019).