

OGC API-Tiles

Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: 2019-03-06

External identifier of this OGC® document: <http://www.opengis.net/doc/{doc-type}/{standard}/{m.n}>

Internal reference number of this OGC® document: YY-nnnrx

Version: 0.0.1

Category: OGC® Implementation Specification

Editor: Joan Masó

OGC API Tiles

Copyright notice

Copyright © 2019 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®ImplementationSpecification
Document subtype: if applicable
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.

Table of Contents

1. Scope	7
1.1. Current scope:	7
2. Conformance	8
3. References	9
4. Terms and Definitions	10
4.1. term name	10
5. Conventions	11
5.1. Identifiers	11
6. Overview	12
6.1. Evolution from OGC Web Services	12
6.2. Tiles and maps	13
6.3. How to approach an OGC API	14
7. Requirement Class "Tiles Core"	17
7.1. Overview	17
7.2. General	18
7.3. API landing page	19
7.3.1. Response	19
7.4. Declaration of conformance classes	19
7.4.1. Response	19
7.5. Collections	20
7.6. Collection	20
7.6.1. Collection Links	21
7.7. Tiles description	21
7.7.1. Operation	21
7.7.2. Response	22
7.8. Tiled data from one collection	26
7.8.1. Operation	26
7.8.2. Parameter tileMatrixSetId	26
7.8.3. Parameter tileMatrix	27
7.8.4. Parameter tileRow	27
7.8.5. Parameter tileCol	28
7.8.6. Response	28
7.8.7. Error conditions	29
8. Requirement Class "Tiles from more than one collection"	30
8.1. Overview	30
8.2. API landing page	30
8.2.1. Response	30
8.3. Declaration of conformance classes	31
8.3.1. Response	31

8.4. Tiles description	32
8.4.1. Operation	32
8.4.2. Response	32
8.5. Tiles from more than one collection	34
8.5.1. Operation	34
8.5.2. Parameter tileMatrixSetId	34
8.5.3. Parameter tileMatrix	35
8.5.4. Parameter tileRow	35
8.5.5. Parameter tileCol	36
8.5.6. Parameter Collections	36
8.5.7. Response	38
8.5.8. Error conditions	38
Annex A: Conformance Class Abstract Test Suite (Normative)	39
A.1. Conformance Class A	39
A.1.1. Requirement 1	39
A.1.2. Requirement 2	39
Annex B: Revision History	40
Annex C: Bibliography	41

i. Abstract

The OGC has started a focused effort to extend their service standards into the Resource Oriented Architecture world. As part of this effort, this standard defines an API for Map Tiles.

The Map Tile API described in this standard builds on the Web Map Tile Service (WMTS) OGC standard. WMTS provides a scalable, high performance services for web based distribution of cartographic maps. WMTS, in turn, complements earlier efforts to develop services for the web based distribution of cartographic maps. In particular, it compliments the OGC Web Map Service (WMS). WMS focuses on rendering custom maps and is an ideal solution for dynamic data or custom styled maps (combined with the OGC Style Layer Descriptor (SLD) standard). WMTS trades the flexibility of custom map rendering for the scalability possible by serving of static data (base maps) where the bounding box and scales have been constrained to discrete tiles. Note that an API version of WMS is also under development.

ii. Keywords

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

ogcdoc, OGC document, tiling, WMTS

iii. Preface

This document defines an OGC standard for a Web Map Tile API standard. A Map Tile enabled API can serve map tiles of spatially referenced data using tile images with predefined content, extent, and resolution. Suggested additions, changes and comments on this standard are welcome and encouraged. Such suggestions may be submitted using the online change request form on OGC web site: http://portal.opengeospatial.org/public_ogc/change_request.php

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.

iv. Submitting organizations

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

Organization name(s)

v. Submitters

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

Name Affiliation

Chapter 1. Scope

This International Standard specifies how to access maps and tiles in a manner independent of the underlying data store through [OpenAPI](<https://www.openapis.org/> [https://www.openapis.org/]). This standard specifies discovery and query operations.

1.1. Current scope:

- Discovery operations allow the API to be interrogated to determine its capabilities and retrieve information (metadata) about this distribution of tiles and maps. This includes the API definition as well as metadata about the feature collections provided through the API and the TileMatrixSets supported by this service.
- Retrieve of maps as defined by the WMS 1.3
- Retrieve of tiles as defined by the WMTS 1.0
- Query about a point in a map or a tile (GetFeatureInfo)
- Retrieve multiple tiles in a single request.

Chapter 2. Conformance

This standard defines **TBD** requirements / conformance classes.

The standardization targets of all conformance classes are "web services".

The main requirements class is:

- **Core.**

The *Core* specifies requirements that all Map Tile APIs have to implement.

TBD requirements classes depend on the *Core* and <enter their purpose here>:

Capture additional requirements classes here

Conformance with this standard 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 to this OGC® interface standard, a software implementation shall choose to implement: * Any one of the conformance levels specified in Annex A (normative). * Any one of the Distributed Computing Platform profiles specified in Annexes TBD through TBD (normative).

All requirements-classes and conformance-classes described in this document are owned by the standard(s) identified.

Chapter 3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC: OGC API (OAPI) Common Specification https://github.com/opengeospatial/oapi_common (in the process of elaboration)

OGC: OGC 17-083r2, OGC Two Dimensional Tile Matrix Set Standard (2019)

In addition, this standard is deeply inspired in concepts defined in the following documents. This standard offers an alternative interface to fulfill similar tasks included in these references.

OGC and ISO: OGC 06-042 1.3.0 OpenGIS Web Map Service (WMS) Implementation Specification

OGC: OGC 07-057, OpenGIS® Web Map Tile Service Implementation Standard (2010)

OGC: OGC 13-082, OGC® Web Map Tile Service (WMTS) Simple Profile (2016)

Chapter 4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

For the purposes of this document, the following additional terms and definitions apply.

4.1. term name

text of the definition

Chapter 5. Conventions

This sections 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.

5.1. Identifiers

The normative provisions in this standard are denoted by the URI

<http://www.opengis.net/spec/{standard}/{m.n}>

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

Chapter 6. Overview

6.1. Evolution from OGC Web Services

OGC Web Service (OWS) standards have historically implemented a Remote-Procedure-Call-over-HTTP architectural style using Extensible Markup Language (XML) for payloads. This was the state-of-the-art when some of the initial versions of OGC Web Services were originally designed in the late 1990s and early 2000s. This architectural style has now a competing RESTful API style that is proposed as an alternative to RPC pattern. A RESTful API style is resource-oriented instead of service-oriented. This OGC API - Maps and Tiles draft specification specifies an API that follows this Web architecture and in particular the W3C/OGC best practices for sharing Spatial Data on the Web as well as the W3C best practices for sharing Data on the Web.

The OGC API – Common draft specification specifies the common kernel of an API approach to services that follows current resource-oriented architecture practices. The draft OGC API - Common specification is the foundation upon which OGC APIs will be built. This common API is to be extended by resource-specific API standards. This draft specification extends OGC API - Common to support Map and Tile resources.

Beside the general alignment with the architecture of the Web (e.g., consistency with HTTP/HTTPS, hypermedia controls), another goal for OGC API standards is modularization. This goal has several facets:

- Clear separation between core requirements and more advanced capabilities. This OGC API – Maps and Tiles draft specification presents the requirements that are relevant for almost everyone who wants to share or use Tiled Map Data on a fine-grained level. Additional capabilities that several communities are using today will be specified as extensions to the Core API.
- Technologies that change more frequently are decoupled and specified in separate modules ("requirements classes" in OGC terminology). This enables, for example, the use/re-use of new encodings for spatial data or API descriptions.
- Modularization is not just about a single "service". OGC APIs will provide building blocks that can be reused in APIs in general. In other words, a server supporting the OGC API - Tiles should not be seen as a standalone service. Rather it should be viewed as a collection of API building blocks which together implement Map and Tile capabilities. A corollary for this is that it should be possible to implement an API that simultaneously conforms to conformance classes from the Feature, Coverage, Map, Tiles, and other future OGC Web API standards.

This approach intends to support two types of client developers:

- Those that have never heard about OGC. Developers should be able to create a client using the API definition without the need to adopt a specific OGC approach (they no longer need to read how to implement a GetCapabilities, allowing them to focus on the geospatial aspects).

- Those that want to write a "generic" client that can access OGC APIs. In other words, they are not specific for a particular API.

As a result of following a RESTful approach, OGC API implementations are not backwards compatible with OWS implementations per se. However, a design goal is to define OGC APIs in a way that an OGC API interface can be mapped to an OWS implementation (where appropriate). OGC APIs are intended to be simpler and more modern, but still an evolution from the previous versions and their implementations making the transition easy (e.g. by initially implementing facades in front of the current OWS services).

This document provides simple examples throughout the document. The examples are based on a dataset that contains buildings and the API provides access to the datasets via a single feature collection ("buildings") and two encodings: JSON and Hypertext Markup Language (HTML).

6.2. Tiles and maps

WMS and WMTS share the concept of a map and the capability to create and distribute maps at a limited resolution and size. In WMS the number of rows and columns can be selected by the user within limits and in WMTS the number of rows and columns of the response is predefined in the tile matrix set.

With time, the concept of a tile has been generalized to other data models such as feature data (some vendors use the expression *vector tiles*) and even to coverage data. This draft specification presents an approach to tiles that can be applied to almost every resource type that returns data representations. If applied in conjunction with the OGC API - Features standard and on top of a feature collection, the expected result is tiled feature data. If applied in conjunction with the OGC API - Maps draft specification and on top of a collection that is transformed into a map by applying a style, the result should be map tiles (usually in PNG or JPEG format).

In this draft specification the OGC API - Tiles is almost fully described. It includes the a core and extensions for defining tile matrix sets, tiles from more that one collection, multi-tiles and multitiles from more than one collection. And info extension is foreseen but not fully developed. In contrast, OGC API - Maps is only partially described based on Testbed-15 requirements. The Maps API is described only to the extent to allow for map tiles to be created on top of a map created by selecting a collection with style or multiple collections with styles. This draft specification contains a section for retrieving a map of an arbitrary number of rows and columns but is not fully formulated. Other extensions for maps are also foreseen. In the future, the WMS SWG could take this document and complete the missing capabilities.

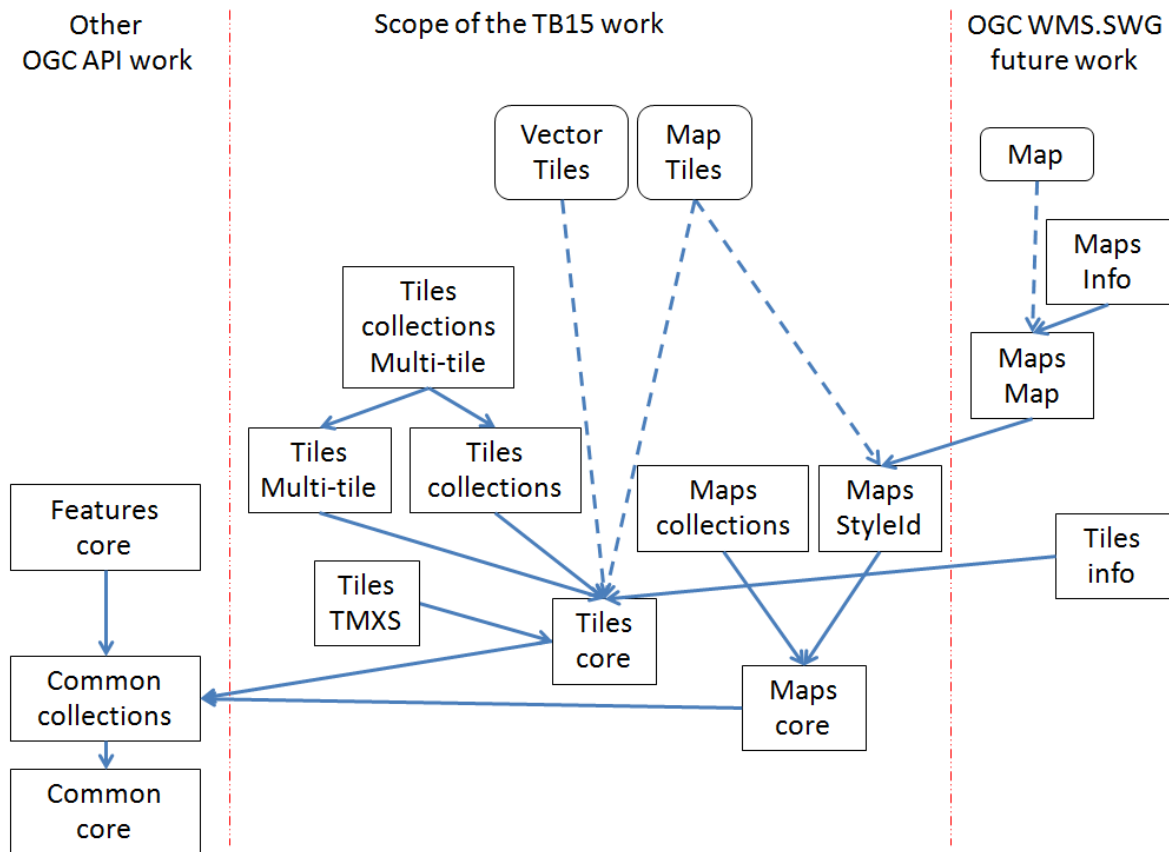


Figure 1. Modular approach in the Maps and Tiles draft specification

6.3. How to approach an OGC API

There are two ways to approach an OGC API.

- Read the landing page, look for links, follow them and discover new links until the desired resource is found
- Read an API definition document that will specify a list of paths to resources.

For the first approach, many resources in the API include links with rel properties to know the reason for this relation. The following figure illustrates does links.

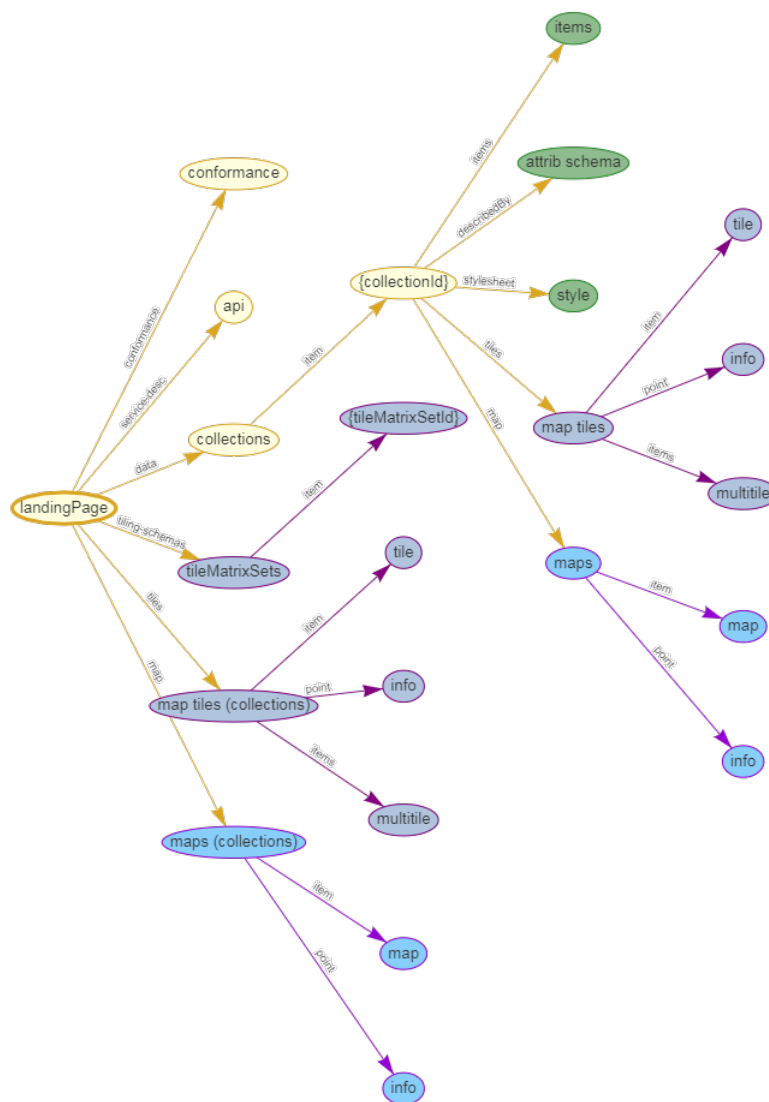


Figure 2. Resources and relations to them via links

For the second approach, the section [\[OpenAPIExamples\]](#) will provide some examples of OpenAPI definition documents that enumerate the paths to get to the necessary resources directly.

Resource name	Common path
Landing page	/
Conformance declaration	/conformance
Collections	/collections
Collection	/collections/{collectionId}
Tiling Schemas	/tileMatrixSets
Tiling Schema	/tileMatrixSets/{tileMatrixSetId}
Tiles	
Vector Tiles description	/collections/{collectionId}/tiles

Resource name	Common path
Vector Tiles description from collections	/tiles
Vector Tile	/collections/{collectionId}/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}
Vector tile collections ¹	/map/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}
Vector Multi-tiles	/collections/{collectionId}/tiles/{tileMatrixSetId}
Vector Multi-tiles collections ¹	/tiles/{tileMatrixSetId}
Map tiles	
Map tiles description	/collections/{collectionId}/map/
Map tiles description collections ¹	/map/tiles
Map tile	/collections/{collectionId}/map/{styleId}/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}
Map tile collections ¹	/map/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}
Map tile multi-tiles	/collections/{collectionId}/map/{styleId}/tiles/{tileMatrixSetId}
Map tile multi-tiles collections ¹	/map/tiles/{tileMatrixSetId}
Maps	
Maps description	/collections/{collectionId}/map
Maps description collections ¹	/map

Table 1. Overview of resources and common direct links defined in the API

¹: In first column of the table, the word "collections" means "from more than one collection"

Chapter 7. Requirement Class "Tiles Core"

7.1. Overview

Requirements Class	
http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core	
Target type	Web API
Dependency	RFC 2616 (HTTP/1.1)
Dependency	RFC 2818 (HTTP over TLS)
Dependency	RFC 3339 (Date and Time on the Internet: Timestamps)
Dependency	RFC 8288 (Web Linking)
Dependency	http://www.opengis.net/spec/tilematrixset/1.0/req/tilematrixset2d
Dependency	http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core
Dependency	http://www.opengis.net/spec/ogcapi-common-1/1.0/req/collections

An API that implements this conformance class provides access to tiled resources of a [dataset](#) [<https://www.w3.org/TR/vocab-dcat/#class-dataset>]. In other words, the API enables the [distribution](#) [<https://www.w3.org/TR/vocab-dcat/#class-distribution>] of that dataset. An implementation of OGC API - Features standard, for example, could be another distribution.

The entry point is a [Landing page](#) (path `/`).

The [Landing page](#) provides links to:

- the [API definition](#) (path `/api`, link relation `service-desc`),
- the [Conformance declaration](#) (path `/conformance`, link relation `conformance`), and
- the [Collections](#) (path `/collections`, link relation `data`).

The [API definition](#) describes the capabilities of the API instance that can be used by clients to retrieve resources from the API or by development tools to support the implementation of API servers and clients. Accessing the [API definition](#) using HTTP GET returns a description of the API.

The [Conformance declaration](#) states the requirements classes from standards or community specifications, identified by a Uniform Resource Identifier (URI), that the API conforms to. Clients can, but are not required to, use this information. Accessing the [Conformance declaration](#) using HTTP GET returns the list of URIs of requirements classes implemented by the API.

The core of the OGC API - Tiles draft specification (as defined in this chapter) does not mandate the inclusion of an explicit definition of any `TileMatrixSet`. This draft specification assumes that

clients and services know about the eight TileMatrixSets defined in OGC 17-083r2 annex D and there is no need to communicate these definitions. An extension to the core provides the capability to include definitions of flexible TileMatrixSets that are explicitly defined.

This draft specification assumes that data is organized into one or more collections. **Collections** provides information about the collections and enumerate the collection identifier.

This document does not specify requirements for collections, and they can consist of features, coverages, a resource that does not represent data per-se (e.g. an annotation) any other resource that can be represented in a tile. collectionId replaces the concept of layer in WMS and WMTS. Maps or tiles can be generated from one collection (or a combination of collections as an extension).

Accessing **Collections** using HTTP GET returns a response that contains at least the list of collections. Accessing **Collections/{collectionId}** using HTTP GET returns a description of a collection with an indication of whether the collection can be retrieved as a map or a tile or both. Accessing the items of a collection is out of the scope of this draft specification but is described in other draft OGC API specifications for features or coverages, for instance. For each **Collection**, a link to metadata about the collection is available (path **/collections/{collectionId}**) with key information about the collection. This information includes:

- A local identifier for the collection that is unique for the dataset;
- An optional title and description for the collection;
- An optional extent that can be used to provide an indication of the spatial and temporal extent of the collection - typically derived from the data;
- A list of TileMatrixSetLink objects relating to the available tiling schemas supported by the collection (from the linked TileMatrixSet member, the client can determine the coordinate reference systems (CRS) in which tiles may be returned by the API)

The **Collection** resource is available at path **/collections/{collectionId}**, often with more details than included in the **Collections** response. In particular, there is a list of links. If there is a link to more metadata about tiles, the collection is available directly as tiles. In the metadata about tiles there are also links and at least one of these links will provide the template to get individual tiles.

7.2. General

Requirement 26	/req/tiles/core/api-common
-----------------------	-----------------------------------

A	An OGC API – Tiles implementation SHALL comply with the requirements specified in the http://www.opengis.net/spec/OAPI_Common/1.0/req/core and http://www.opengis.net/spec/OAPI_Common/1.0/req/collections Requirements Classes of the OGC API-Common version 1.0 Standard.
---	--

In practice, this means that the landing page and the conformance page follow OGC API - Common core and collections requirement classes. This draft specification provides extra additions to the OGC API - Common requirements that are particular to tiles.

7.3. API landing page

The landing page provides links to start exploring the resources offered by the API. The landing page mainly consists of a list of links. OGC API - Common already requires some common links that are enough for this draft specification core.

7.3.1. Response

There are no required variations to the landing page.

7.4. Declaration of conformance classes

To support "generic" clients that want to access multiple OGC API standards and extensions - and not "just" a specific API / server, the API has to declare the requirements classes it implements and conforms to.

7.4.1. Response

The conformance page mainly consists of a list of links. OGC API - Common already requires some links.

Requirement 27	/req/tiles/core/conformance-success
A	The API conformance path SHALL advertise the tiles core conformance class as links to http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core .

In the conformance page (typically in JSON format) the links follow the link schema defined in the OGC API - Common. Below is an example fragment of the response to an OGC API - Tiles conformance information page.

```
{
  "conformsTo": [
    "http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core",
    "http://www.opengis.net/spec/ogcapi-common-1/1.0/req/collections",
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core"
  ]
}
```

7.5. Collections

This draft specification includes dependencies on OGC API - Common collections. Collections are mandatory in the core of this draft specification because collections are the object that will be included in a tile.

Collections will enumerate the collectionId identifiers available in this implementation of the OGC API draft specification as well as basic information about each collectionId: id, title, description, extent, CRS and links. This common response is considered enough for a general description of the collection.

Requirement 28	/req/tiles/core/tc-md-collection-links
A	In the array of collections included in the response, for each collection that can be retrieved as tiles, a links property of the collection SHALL include a link to the description of the collection (rel: self) (in addition to other links specified in OGC API Commons). This link will allow to request information about this collection and, in particular to get another link, this time to the information about the tile structure.

More specific details about the collection can be found following the link to the individual collections that follow the pattern /collections/{collectionId}

NOTE

The collectionId substitutes the concept of "layer" in WMTS 1.0.

7.6. Collection

This draft specification includes dependencies on the OGC API - Common collection requirement. The response to the operation is extended with a new link for the tiles description.

7.6.1. Collection Links

Requirement 29	/req/tiles/core/tc-tile-desc-links
A	A links property of the collection SHALL include a link to the description of the tiles (rel: tiles) (in addition to other links specified in OGC API Commons).

Example 2. Fragment of the collection links.

```
"id": "buildings",
"title": "Buildings in the city of Bonn",
"description": "This collection contains buildings",
"attribution": "OpenStreetMap",
"extent": {
  ...
}
"links": [
  ...
  {
    "href": "http://data.example.com/collections/buildings/tiles",
    "rel": "tiles",
    "type": "application/json",
  }
]
```

7.7. Tiles description

The response to this operation contains the necessary metadata to enable a client application to formulate a tile request from a single collection.

7.7.1. Operation

Requirement 30	/req/tiles/core/sct-op
----------------	------------------------

A	Every resource available as tiles SHALL support an operation to retrieve the description of the tiles the API implementation can provide, available as a HTTP GET request to a URI that will be composed by two parts: the initial part is the URI of a resource that can be represented as tiles and the final part follows the pattern /tiles. Only the resources or collection that supports this operation can be retrieved as tiles.
---	---

The request of this operation has no parameters.

7.7.2. Response

A successful response to a tiles request for a collection that can be retrieved as tiles will respond with a data structure with specific information necessary to get tiles representing the resource collection. In this core draft specification, the response is only required to inform about from which tile matrix sets tiles can be retrieved and the URL template to a tile.

Requirement 31	/req/tiles/core/sct-tmxslink
-----------------------	-------------------------------------

A

The content of the response to a successful execution SHALL contain a property called *tileMatrixSetLinks* with a list of *tileMatrixSetLink* objects following a data model defined in the clause 7.3 of OGC 17-083r2. In the core specification *tileMatrixSetLink* is only used for referencing the supported TileMatrixSets for the tiles limiting it to the following schema (expressed as an OpenAPI Specification 3.0 fragment):

```
tileMatrixSetLink-set:
  description: This list of tileMatrixSetLink
objects, as defined in OGC 17-083r2 supported by
this collectionId.
  type: array
  items:
    $ref:
'#/components/schemas/tileMatrixSetLink-entry'
  tileMatrixSetLink-entry:
    type: object
    required:
      - tileMatrixSet
    properties:
      tileMatrixSet:
        type: string
        example: 'WebMercatorQuad'
      tileMatrixSetURI:
        type: string
        format: uri
        example:
'http://www.opengis.net/def/tilematrixset/OGC/1.0
/WebMercatorQuad'
```


Example 3. Example of a tiles metadata response.

```
{
  "tileMatrixSetLinks": [
    {
      "tileMatrixSet": "WorldMercatorWGS84Quad",
      "tileMatrixSetURI":
"http://schemas.opengis.net/tms/1.0/json/examples/WorldMercatorWGS84Quad.json"
    }
  ],
  ...
  "links": [
    ...
    {
      "href":
"http://data.example.com/collections/buildings/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}.png",
      "rel": "item",
      "type": "image/png",
    }
    ...
  ]
}
```

Recommendation 32	/rec/tiles/core/sct-tmxslink
A	This core requirements class does not provide any mechanism to defined TileMatrixSets so if this mechanism is not provided in an extension, the tileMatrixSetURI SHOULD point to one of the 8 URIs defined in the OGC 17-083r2 Annex D.
B	The server SHOULD do a effort to provide to the client a way to get full description of the TileMatrixSet. Even if the TileMatrixSet is not directly defined by the API, when a full definition of the TileMatrixSet is available as a resolvable URL, a resolvable URL SHOULD be used as the value of the tileMatrixSetURI.

Resolvable URLs for the 8 URIs defined in the OGC 17-083r2 Annex D are available in the OGC schemas repository in XML, JSON and RDF formats. For example, JSON descriptions can be found here: <http://schemas.opengis.net/tms/1.0/json/examples/>

Requirement 33	/req/tiles/core/sct-tile-examples
A	The content of the response to a successful execution SHALL include at least a link to a tile URI template (rel: item).
B	These links SHALL provide a URL template with the fragment /tiles followed by the variables {tileMatrixSetId}, {tileMatrix}, {tileRow} and {tileCol}. Once the variables are substituted by their respective valid values, a URL to a tile is obtained.
C	There SHALL be a link to a tile URI template for each format that the server supports (the format is indicated in the 'type' attribute of the link)

One common order used in URL templates for tiles is ... /tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}, but this draft specification allows for other URL template composition.

URL template variable	Meaning	Possible values
TileMatrixSetId	tile matrix set identifier	One of the identifiers included in Annex D of OGC 17-083r2 or an identifier defined by extensions of this core
TileMatrix	tile matrix identifier	Identifier of the tile matrix (representing a zoom level, a.k.a. a scale) listed in the TileMatrixSet definition
TileRow	row index of tile matrix	A non-negative integer between 0 and the MatrixHeight – 1. If there is a TileMatrixSetLimits the value is limited between MinTileRow and MaxTileRow
TileCol	column index of tile matrix	A non-negative integer between 0 and the MatrixWidth – 1. If there is a TileMatrixSetLimits the value is limited between MinTileCol and MaxTileCol

Table 2. URI template variables for tiles and valid values

```
links:
[
  {
    "href":
"http://data.example.com/collections/buildings/tiles/{tileMatrixSetId}/
{tileMatrix}/{tileRow}/{tileCol}",
    "rel": "tiles",
    "type": "image/png",
  }
]
```

7.8. Tiled data from one collection

The core of the OGC API -Tiles draft specification provides a mechanism to select and retrieve a tile in a TileMatrixSet. If the service does not advertise any other TileMatrixSet (this core does not describe any mechanism to do that, but an extension will do it) only the TileMatrixSet identifiers specified in the Annex D.1 of the OGC 17-083r2 standard can be used.

7.8.1. Operation

Requirement 34	/req/tiles/core/tc-op
A	Every tile SHALL be available as a HTTP GET request to a URI that will be composed by two parts: The first part is the URI of a resource that can be represented as tiles and the second part follows the pattern /tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}

Typical resources that can be retrieved as tiles are: features (/collections/{collectionId}), coverages (/collections/{collectionId}/coverage/{coverageId} or /coverage/{coverageId}) or maps (/collections/{collectionId}/map/styleId).

NOTE | The common path for coverages is still under discussion.

7.8.2. Parameter tileMatrixSetId

Requirement 35	/req/tiles/core/tc-tilematrixsetid-definition
----------------	---

A	<p>The operation SHALL support a parameter <code>tileMatrixSetId</code> with the following characteristics (shown as OpenAPI Specification 3.0 fragment):</p> <pre> name: tileMatrixSetId in: path description: Identifier of a specific tiling scheme. It can be one of those specified in Annex D.1 of the OGC 17-083r2 standard or one defined in this service. required: true schema: type: string example: WebMercatorQuad </pre>
---	--

7.8.3. Parameter `tileMatrix`

Requirement 36	<code>/req/tiles/core/tc-tilematrix-definition</code>
A	<p>The operation SHALL support a parameter <code>tileMatrix</code> with the following characteristics (shown as OpenAPI Specification 3.0 fragment):</p> <pre> name: tileMatrix in: path description: Identifier selecting one of the scales defined in the TileMatrixSet and representing the scaleDenominator the tile. required: true schema: type: string example: '11' </pre>

7.8.4. Parameter `tileRow`

Requirement 37	<code>/req/tiles/core/tc-tilerow-definition</code>
----------------	--

A	<p>The operation SHALL support a parameter tileRow with the following characteristics (shown as OpenAPI Specification 3.0 fragment):</p> <pre> name: tileRow in: path description: Row index of the tile on the selected TileMatrix. It cannot exceed the MatrixWidth-1 for the selected TileMatrix required: true schema: type: integer minimum: 0 example: '827' </pre>
---	--

7.8.5. Parameter tileCol

Requirement 38	/req/tiles/core/tc-tilecol-definition
A	<p>The operation SHALL support a parameter tileCol with the following characteristics (shown as OpenAPI Specification 3.0 fragment):</p> <pre> name: tileCol in: path description: Column index of the tile on the selected TileMatrix. It cannot exceed the MatrixHeight-1 for the selected TileMatrix. required: true schema: type: integer minimum: 0 example: 1231 </pre>

7.8.6. Response

A successful response to a tile request will be consistent with the media type of resource requested. This draft specification does not impose any media type. For example:

- For features the media type can be geojson or Mapbox vector tiles;

- For coverages the response may be a geotiff;
- For maps the response may be a jpeg or a png.

Requirement 39	/req/tiles/core/tc-success
A	A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 .
B	The content of that response SHALL be consistent with the format requested and represent elements inside or intersecting with the spatial extent of the geographical area of the tile identified by TileMatrixSet, TileMatrix, TileRow and TileCol.

Normally, the content partially outside the tile bounding box will be clipped and this is particularly true when tiles are in raster format. Nevertheless, tiles containing features in vector format may not clip features that are partially outside.

Recommendation 3	/rec/tiles/core/tc-success-scale
A	The content of that response should be simplified to comply with the scale denominator represented by the TileMatrix identified. Full resolution geographical elements will only be provided for the lower values of scale denominators.

7.8.7. Error conditions

A general summary of the HTTP status codes can be found in the OGC API - Common.

If the parameter value **tileMatrixSetId** is not available by the server for this resource or the parameters values **tileMatrix**, **tileRow**, **tileCol** are out-of-range, the status code of the response will be 404.

Chapter 8. Requirement Class "Tiles from more than one collection"

8.1. Overview

Requirements Class	
http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/collections	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core

In previous clauses tiles that are produced from one, and only one resource were discussed. This scenario is achieved by concatenating the tile path to a resource (e.g. a feature collection). This requirements class is an extension of the core requirements class that defines how to create tiles that combine more than one resource. This is achieved by having the tile path also available at the root of the service.

8.2. API landing page

The landing page provides links to start exploring the resources offered by the API. It mainly consists of a list of links. The core of this draft specification does not add anything to the links required by OGC API - Common. This requirements class for *tiles from more than one collection* requires a new link for getting the description of the *tiles from more than one collection* on top of the common ones.

8.2.1. Response

Requirement 40	/req/tiles/collections/root-success
A	The API SHALL advertise a URI to retrieve tiles definitions defined by this service as links to the descriptions paths with rel: tiles .

In the landing page, in JSON format, the links follow the link schema defined in the OGC API - Common. Below is an example fragment of the response to an OGC API - Tiles landing page showing the new link.

```
{
  links: [
    ...,
    {
      "href": "http://data.example.org/tiles",
      "rel": "tiles",
      "type": "application/json",
      "title": "Link to information on map tiles combining more than
one collection",
    }
  ]
}
```

8.3. Declaration of conformance classes

8.3.1. Response

The conformance page mainly consists of a list of links. OGC API - Common already requires some links.

Requirement 41	/req/tiles/collections/conformance-success
A	The API conformance path SHALL advertise the capability of generating tiles from multiple collections adding the conformance class for this capability as a link to http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/collections .

On the conformance page (typically in JSON format) the links follow the link schema defined in the OGC API – Common draft specification. The following is an example fragment from the response to an OGC API - Tiles conformance information page showing the support for *tiles from more than one collection*


```
{
  "conformsTo": [
    "http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core",
    "http://www.opengis.net/spec/ogcapi-common-1/1.0/req/collections",
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core"
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/collections"
  ]
}
```

8.4. Tiles description

The response to the tiles description operation contains the necessary information to later formulate a tile request of tiles from more than one collection.

8.4.1. Operation

Requirement 42	/req/tiles/collections/ts-op
A	The server SHALL support an operation to retrieve the description of the tiles from more than one collection, available as a HTTP GET request to a URI that is composed by two parts: the first part is the URI of a resource that can be represented as tiles (e.g. /map or simply /) and the second part follows the pattern /tiles.

The request of this operation has no parameters.

8.4.2. Response

A successful response to a tiles request for more than one collection will respond with a data structure with specific information necessary to get tiles representing the resource collection. In this requirements class, the response only provides the URL template to retrieve a tile.

Requirement 43	/req/tiles/collections/ts-tile-examples
A	The content of the response to a successful execution SHALL include at least one link to a tile URI template (rel: item).

B	These links SHALL provide a URL template with the fragment /tiles followed by the variables {tileMatrixSetId}, {tileMatrix}, {tileRow} and {tileCol}. Once the variables are substituted by their respective valid values, a URL to a tile is obtained.
C	There SHALL be a link to a tile URI template for each format that the server supports (the format is indicated in the 'type' attribute of the link)

One common order used in URL templates for tiles is: /tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}. However, this draft specification allows for other URL template composition.

URL template variable	Meaning	Possible values
TileMatrixSetId	tile matrix set identifier	The identifiers included in Annex D of OGC 17-083r2 or defined by extensions of the core requirements class.
TileMatrix	tile matrix identifier	Identifier of the tile matrix (representing a zoom level, a.k.a. a scale) listed in the TileMatrixSet definition
TileRow	row index of tile matrix	A non-negative integer between 0 and the MatrixHeight – 1. If there is a TileMatrixSetLimits the value is limited between MinTileRow and MaxTileRow
TileCol	column index of tile matrix	A non-negative integer between 0 and the MatrixWidth – 1. If there is a TileMatrixSetLimits the value is limited between MinTileCol and MaxTileCol

Table 3. URI template variables for tiles and possible values

```
links:
[
  {
    "href":
"http://data.example.com/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}
/{tileCol}",
    "rel": "item",
    "type": "image/png",
  }
]
```

In general, the `tileMatrixSetLinks` and the `tileMatrixSetLimits` can be determined by examining this information in the individual collections. In some cases, the server could also include the `tileMatrixSetLinks` data structure as part of the response to this operation. Clients should be prepared to determine if a `tileMatrixSetLinks` data structure is not provided in certain combinations of collections by examining the `tileMatrixSet` values and limits from the information in the individual collections and calculating the limits as the most restrictive intersection of them.

8.5. Tiles from more than one collection

This operation allows retrieving a single tile that represents information coming from more than one collection.

8.5.1. Operation

Requirement 44	/req/tiles/collections/tcs-op
A	The server SHALL support the HTTP GET operation at the path <code>/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}</code>

8.5.2. Parameter `tileMatrixSetId`

Requirement 45	/req/tiles/collections/tcs-tilematrixsetid-definition
-----------------------	--

A	<p>The operation SHALL support a parameter <code>tileMatrixSetId</code> with the following characteristics (shown as OpenAPI Specification 3.0 fragment):</p> <pre> name: tileMatrixSetId in: path description: Identifier of a specific tiling scheme. It can be one of the specified in Annex D.1 of the OGC 17-083r2 standard or one defined in this service. required: true schema: type: string example: WebMercatorQuad </pre>
---	--

8.5.3. Parameter `tileMatrix`

Requirement 46	<code>/req/tiles/collections/tcs-tilematrix-definition</code>
A	<p>The operation SHALL support a parameter <code>tileMatrix</code> with the following characteristics (shown as OpenAPI Specification 3.0 fragment):</p> <pre> name: tileMatrix in: path description: Identifier selecting one of the scales defined in the TileMatrixSet and representing the scaleDenominator the tile. required: true schema: type: string example: '11' </pre>

8.5.4. Parameter `tileRow`

Requirement 47	<code>/req/tiles/collections/tcs-tilerow-definition</code>
----------------	--

A	<p>The operation SHALL support a parameter tileRow with the following characteristics (shown as OpenAPI Specification 3.0 fragment):</p> <pre> name: tileRow in: path description: Row index of the tile on the selected TileMatrix. It cannot exceed the MatrixWidth-1 for the selected TileMatrix required: true schema: type: integer minimum: 0 example: '827' </pre>
---	--

8.5.5. Parameter tileCol

Requirement 48	/req/tiles/collections/tcs-tilecol-definition
A	<p>The operation SHALL support a parameter tileCol with the following characteristics (shown as OpenAPI Specification 3.0 fragment):</p> <pre> name: tileCol in: path description: Column index of the tile on the selected TileMatrix. It cannot exceed the MatrixHeight-1 for the selected TileMatrix. required: true schema: type: integer minimum: 0 example: 1231 </pre>

8.5.6. Parameter Collections

Requirement 49	/req/tiles/collections/tcs-collections-definition
----------------	---

A	<p>The operation SHALL support an optional parameter collections with the following characteristics (shown as OpenAPI Specification 3.0 fragment)</p> <pre> name: collections in: query required: false style: form explode: false schema: type: array items: type: string </pre>
B	The parameter collections SHALL contain a comma-separated list of collection identifiers.
C	Only the collections that advertise a link type=tiles in the /collections/{collectionId} SHALL be included.
D	Only the collections that support the same TileMatrixSetId parameter value SHALL be included.

Recommendation 4 /rec/tiles/collections/tcs-collections-definition	
A	If the parameter collections is missing, and when it is possible and sensible, all collections supporting the TileMatrixSetId parameter value SHOULD be represented in the tiles.
B	The collection ids that can be used for this operation SHOULD be listed in the description of the collections parameter in the API definition

Permission 2	/per/tiles/collections/tcs-collections-definition
---------------------	--

A	If the parameter collections is missing and if it is not possible and sensible to represent all collections in tiles (e.g. it compromises performance or tiles are become packed with too many elements), the server is allowed to select only the most significant collections.
---	---

8.5.7. Response

A successful response to a tile request is consistent with the media type of resource requested. This draft specification does not impose any media type. For example, for features the media type can be GeoJSON or Mapbox vector tiles, for coverages it may be a GeoTIFF, and for maps it may be a JPEG or a PNG.

Requirement 50	/req/tiles/collections/tcs-success
A	A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 .
B	The content of that response SHALL be consistent with the format requested and represent elements inside or intersecting with the spatial extent of the geographical area of the tile identified by TileMatrixSet, TileMatrix, TileRow and TileCol.
C	The content of that response SHALL be simplified to comply with the scale denominator represented by the TileMatrix identified. Full resolution geographical elements will only be provided for the lower values of scale denominators.

8.5.8. Error conditions

If the value of the parameter **tileMatrixSetId** is not available by the server for this resource or the values of the parameters **tileMatrix**, **tileRow**, **tileCol** are out-of-range, the status code of the response is 404.

If the value of the parameter **collections** contains a collection id that does not exist on the server, the status code of the response is 404.

If the value of the parameter **collections** has a wrong format or combines collections and some of them are not compatible with the **tileMatrixSetId** value, the status code of the response is 500.

Annex A: Conformance Class Abstract Test Suite (Normative)

NOTE

Ensure that there is a conformance class for each requirements class and a test for each requirement (identified by requirement name and number)

A.1. Conformance Class A

A.1.1. Requirement 1

Test id:	/conf/conf-class-a/req-name-1
Requirement:	/req/req-class-a/req-name-1
Test purpose:	Verify that...
Test method:	Inspect...

A.1.2. Requirement 2

Annex B: Revision History

Date	Release	Editor	Primary clauses modified	Description
2019-03-21	Template	C. Heazel	all	initial template

Annex C: Bibliography

- W3C/OGC: Spatial Data on the Web Best Practices, W3C Working Group Note 28 September 2017, <https://www.w3.org/TR/sdw-bp/>
- W3C: Data on the Web Best Practices, W3C Recommendation 31 January 2017, <https://www.w3.org/TR/dwbp/>
- W3C: Data Catalog Vocabulary, W3C Recommendation 16 January 2014, <https://www.w3.org/TR/vocab-dcat/>
- IANA: Link Relation Types, <https://www.iana.org/assignments/link-relations/link-relations.xml>