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

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

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>:

C**apture 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 and 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. geospatial resource

A resource that consists in a set of geospatial data

4.2. geospatial representation

An resources that represents an aspect or data model (e.g. feature items, tiles, metadata, schemas,…) of a more generic geospatial resource (e.g. a collection)

NOTE

Do not confuse this with a resource representation. While resource representations share the same path and are selected by format negotiation, geospatial representations use different paths. Commonly a geospatial representations is a child path of a geospatial resource

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 Tiles draft specification presents the requirements that are relevant for almost everyone who wants to share or use Tiled 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 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).

This draft specification only describes the core capabilities for the API - Tiles. Other extensions to the core will define how to add tile matrix set descriptions, multi-tiles and simple pixel queries. To produce map tiles, some modules of the OGC API - Maps will be need.

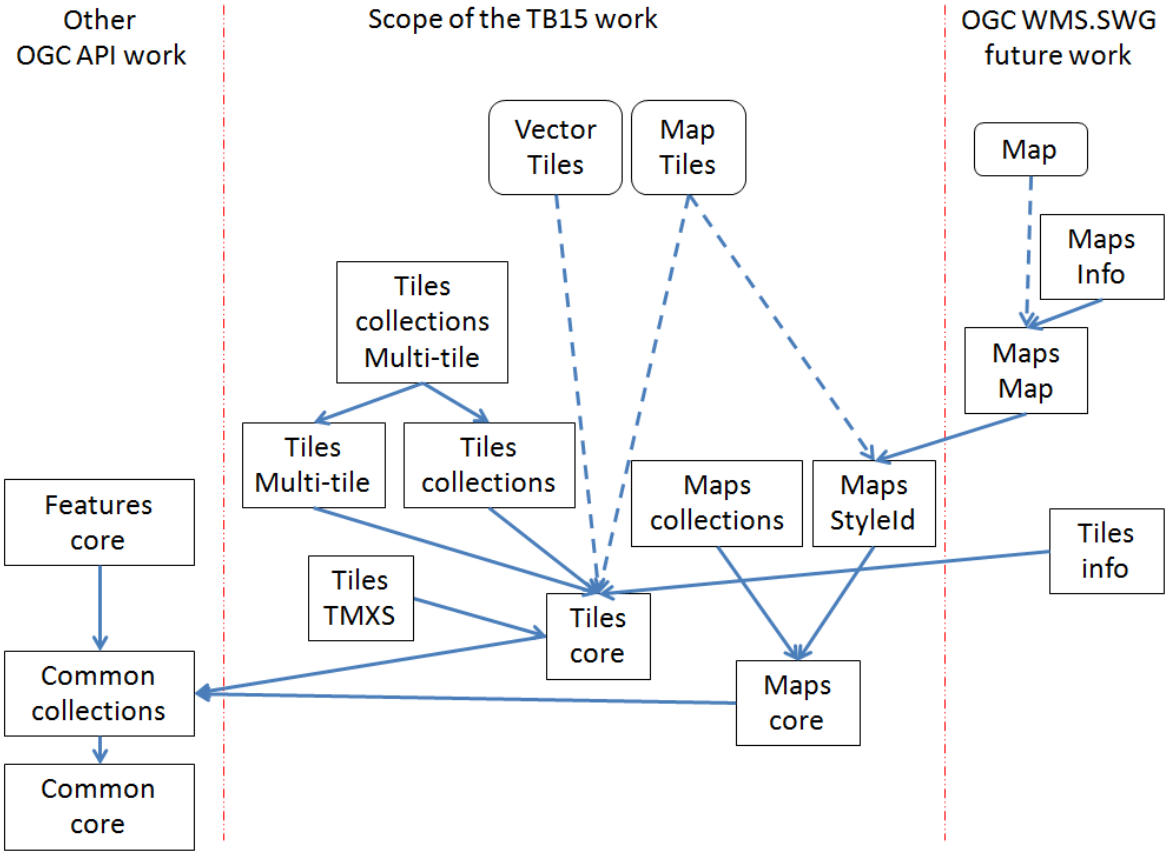

Figure 1. Modular approach in the Maps and Tiles draft specification

6.3. How to approach an OGC API

This specification cannot be implemented alone and should be considered a building block that could be applied to one or more existing resources in the API to get access spatial subsets of existing resources. This core defined two ways to get tiles defined in to separated conformance classes. Developers are free to implement one of the approaches or both:

- tiles are applied as a transformation of a resource to obtain another resource as tiles.
- tiles are resources that are the result to combine one or more resources that characterized by their URLS.

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

This is a building block for the an OGC API that is able to provide geospatial resources. When applying the building block to a geospatial resource, it becomes available as tiles. The server can select which resources are available as tiles and will advertise which resources are available as tiles.

This building block does not specify how to get an API definition, the conformance class list or the geospatial resources lists. The standard assumes that the first two are defined by an API specification (e.g. OGC API Common) and the later by an OGC API for geospatial resource (e.g. OGC API Features).

The core of the OGC API - Tiles core draft specification 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 (or compatible future update of it) and there is no need to define new TileMatrixSets. 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 in on or more geospatial resources (e.g. the "collections" in OGC API features). Geospatial resources are referenced using URIs.

This document does not specify any requirements for geospatial resources, and they could be features, coverages, a resource that does not represent data per-se (e.g. an annotation) as far as they can be represented a tile. The resource path replaces the concept of layer in WMS and WMTS. In this core tiles can be generated from only one geospatial resource (tiles that are generated as a combination of geospatial resources will be defined as an extension).

Accessing the geospatial resource content (other than as tiles) or its descriptions is out of the scope of this draft specification. If a description of the geospatial resource exists and it has a

mechanism to add links to it, this specification will indicate the need to add a link to the tile representation description.

The tile representation description will include metadata about tiles as well as links to other resources including at least one with a template to get individual tiles.

7.2. General

Recommendation 5	**/rec/tiles/core/api-common**
A	An implementation this standard should consider to implement 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.

This building block stays flexible and does not require to implement OWS Common, allowing for other API architectures outside the OGC API framework to adopt it. However, a server under the OGC APIs is expected to implement OGC API Common. If so, in practice, this means that the landing page and the conformance page follow OGC API - Common core and collections requirement classes will be used. Temporarily, it is also possible to combine this building block with OGC API features v1 that is not tied to OWS Common.

7.3. 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.

7.3.1. Response

The conformance page mainly consists of a list of links.

Requirement 23	**/req/tiles/core/conformance-success**
A	If the API has a mechanism to advertise conformance classes, the API SHALL advertise the tiles core conformance class with a link to http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core.

If the server declares also conformity to OGC API - Common or to OGC API - Features v1, then it has to consider the OGC API - Common requirements for declaring conformance, i.e. the use of a

the conformance page. In the JSON format the conformance page is an array of links following the link schema defined in the OGC API - Common or in OGC API - Features v1. Below is an example fragment of a conformance information page of an API conformant to OGC API - Common and OGC API - Tiles.

Example 1. Conformance Information Page fragment

```
{
  "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"
  ]
}
```

7.4. Geospatial resources

This draft specification does not specify how geospatial resources are exposed in the API and if they have the possibility to be retrievable as geospatial data (e.g. feature items). For example OGC API features v1 includes the definition of collections and each collection is available in the /collections/{collectionId} path. OGC API common will provide a similar mechanism. Other paths in the API could also give access to geospatial resources.

NOTE	The concept of geospatial resource path substitutes the concept of "layer" in WMTS 1.0 but it intended to give a better integration between data visualization and data access.

7.5. Tiles description

A tile description contains the necessary metadata to enable a client application to formulate a tile request from a single geospatial resource.

7.5.1. Tiles description path

Requirement 24	**/req/tiles/core/sct-op**
A	Every geospatial resource available as tiles SHALL support an path URL and a HTTP GET operation to retrieve the description of the tiles the API implementation can provide

B	The URI shall be composed by two parts: the initial part is the URI of the geospatial resource that can be represented as tiles and the final part follows the pattern `/tiles`

This standard does not specify the need for any additional query parameter in the GET request.

7.5.2. Tiles description Link

Requirement 25	**/req/tiles/core/tc-tile-desc-links**
A	If the API has a mechanism for their geospatial resources to expose links to geospatial representations (e.g. feature items), the API SHALL include a `link` with the `href` pointing to a the description of the tiles that presents a tile representation of this geospatial resource and with `rel: "tiles"`.

For example, the OGC API Features v1 return a list of links that include geospatial representations for each geospatial resource in the /collections/{collectionId} path. OGC API Common is expected to provide a similar mechanism. In the JSON response, the array `links` is the place for adding a resource reference to the 'tiles' description.

Example 2. Fragment of a collection with a links array with one item of the array pointing to a tiles description.

```
  "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",
    }
  ]
}
```

7.5.3. Response

A successful GET response to a tiles description resource will respond a data structure with the specific information necessary to build a complete GET request to the tiles representing the geospatial resource. 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 the tiles.

Requirement 26	**/req/tiles/core/sct-tmxslink**

A	If the tiles are available in a tile matrix set different from WebMercatorQuad, the content of the response to a successful execution to a tiles description 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'`

Recommendation 6	/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 27	/req/tiles/core/sct-tile-examples
A	The content of the response to a successful execution of a tile description 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 file format that the server supports (the format is indicated in the 'type' attribute of the link)
D	A property `templated` SHALL be part of the link properties to indicate that the link needs to be processed to substitute the templated variables with valid values before being used as a URL to a tile.

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

NOTE The geospatial resource URL is expected to be the first part of the URL template (presented by the '...' in the previous note) but this standard does not mandate this.

Example 3. Example of a tiles metadata response.

```
{
  "tileMatrixSetLinks": [
    {
      "tileMatrixSet": "WorldMercatorWGS84Quad",
      "tileMatrixSetURI":
"http://schemas.opengis.net/tms/1.0/json/examples/WorldMercatorWGS84Qua
d.json"
    }
  ],
  ...
  "links": [
    ...
    {
     "href":
"http://data.example.com/collections/buildings/tiles/{tileMatrixSetId}/
{tileMatrix}/{tileRow}/{tileCol}.png",
     "templated": true,
     "rel": "item",
     "type": "image/png",
   }
   ...
  ]
}
```

NOTE The use of "templated" is inspired in the JSON Hypertext Application Language (HAL), https://tools.ietf.org/html/draft-kelly-json-hal-08

The following table explains the meaning to the URI template variables.

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

URL template variable	Meaning	Possible values
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 1. URI template variables for tiles and valid values

7.6. A tile from a geospatial resource

A tile resource is a geospatial representation of a fragment of a geospatial resource that is spatially constrained tot the boundaries of the selected tile in a tile matrix set.

7.6.1. Tile path and link

As described before a tile path is obtained by extracting a tile URL templated from one of the links with `rel`: `item` in tiles description document and substituting the templated variables of with valid values.

7.6.2. Operation

Requirement 28	/req/tiles/core/tc-op
A	A tile that contains available data 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 geospatial resource that can be represented as tiles and the second part follows the pattern /tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}

Typical geospatial resources that can be retrieved as tiles are: features (in OGC API Features v1 represented by /collections/{collectionId}), or full maps (specified in the OGC API Maps).

7.6.3. Parameter tileMatrixSetId

Requirement 29	/req/tiles/core/tc-tilematrixsetid-definition

A	The operation SHALL support a parameter `tileMatrixSetId` with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
	
	`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`

The core of the OGC API -Tiles standard 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) the TileMatrixSet identifiers possible are limited to the ones specified in the Annex D.1 of the OGC 17-083r2 standard.

7.6.4. Parameter tileMatrix

Requirement 30	/req/tiles/core/tc-tilematrix-definition

A	The operation SHALL support a parameter `tileMatrix` with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
	
	`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'`

7.6.5. Parameter tileRow

Requirement 31	/req/tiles/core/tc-tilerow-definition

A	The operation SHALL support a parameter `tileRow` with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
	
	`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'`

7.6.6. Parameter tileCol

Requirement 32	/req/tiles/core/tc-tilecol-definition

A	The operation SHALL support a parameter `tileCol` with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
	
	`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`

7.6.7. Response

A successful response to a tile GET operation will be consistent with the media type of resource requested. This draft specification does not impose any media type or file format. 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 33	**/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. This is particularly true and easy to when tiles are in raster format (e.g. map tiles). Nevertheless, tiles containing features in vector format may not clip features that are partially outside to ensure continuity of features of for preformance.

Recommendation 7	**/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 are only expected for the lower values of scale denominators.

7.6.8. Error conditions

A general summary of the HTTP status codes can be found in OGC API - Features v1 and in 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, of the tile ins not provided due to lack of data in the area, the status code of the response will be `404`.

Chapter 8. Requirement Class "Tiles root"

8.1. Overview

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

In previous clauses tiles are produced from one, and only one geospatial resource available in this API. This scenario is achieved by concatenating the tile path to the geospatial resource.

This OGC API requirements class is an extension of the core requirements class that defines how to create tiles that combine one or more geospatial resources in any way that is decided in the client side. This is achieved by having the tile path available at the root of the service.

It has been argued that this approach is too flexible. In an API that has several geospatial resources, the number of potential combinations of geospatial resources may be too big to be efficiently handle. If the implementers see a potential performance issue, they may not choose to declare conformity to this requirements class.

8.2. General

This building block stays flexible and does not require to implement OWS Common, allowing for other API architectures outside the OGC API framework to adopt it. However, a server under the OGC APIs is expected to implement OGC API Common. If so, in practice, this means that the landing page and the conformance page follow OGC API - Common core. Temporarily, it is also possible to combine this building block with OGC API features v1 that is not tied to OWS Common.

8.3. 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 to root resources. This standard extension requires a new link in the landing page for getting a description of the URL that allows for retrieving tiles of one or more resources.

8.3.1. Response

Requirement 34	/req/tiles/root/root-success

A	If the API has mechanism to expose root resources (e.g. a landing page), the API SHALL advertise a URI to retrieve tile 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 or in OGC API - Features v1. Below you can find an example fragment of the response to an OGC API - Tiles landing page showing the link to root tiles.

Example 4. API Landing Page fragment that advertises the path to get tiles for more than one collection

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

8.4. Declaration of conformance classes

8.4.1. Response

The conformance page mainly consists of a list of links.

Requirement 35	**/req/tiles/root/conformance-success**
A	If the API has a mechanism to advertise conformance classes, the API 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/root.

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*

Example 5. Conformance Information Page fragment

```
{
  "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/root"
  ]
}
```

8.5. Root 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.

8.5.1. Operation

Requirement 36	**/req/tiles/toot/ts-op**
A	The server SHALL support an operation to retrieve the description of the root tiles 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.

8.5.2. Response

A successful response to a tiles request for a root tiles will return a data structure with a link to get tiles representing the resources and other relevant resources. This requirements class, the response only requires a URL template to retrieve a tile.

Requirement 37	**/req/tiles/root/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 composed by the URL of this resource 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 2. URI template variables for tiles and possible values

Example 6. API tiles response fragment with the link to retrieve tiles

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

In general, the `tileMatrixSetLinks` and the `tileMatrixSetLimits` can be determined by examining this information in the individual geospatial resource `tiles` of geospatial resource. 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 geospatial resources by examining the tileMatrixSet values and limits from the information in the individual geospatial resources and calculating the limits as the most restrictive intersection of them.

8.6. Tiles

This operation allows retrieving a single tile that represents information coming from one or more geospatial resources.

8.6.1. Operation

Requirement 38	**/req/tiles/root/tcs-op**
A	The server SHALL support a set of HTTP GET operations following a URL template composed by the the root tile resource URL followed by values that substitute the variables {tileMatrixSetId}, {tileMatrix}, {tileRow} and {tileCol}.

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.

8.6.2. Parameter tileMatrixSetId

Requirement 39	/req/tiles/root/tcs-tilematrixsetid-definition

A	The operation SHALL support a parameter `tileMatrixSetId` with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
	
	`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`

8.6.3. Parameter tileMatrix

Requirement 40	/req/tiles/root/tcs-tilematrix-definition

A	The operation SHALL support a parameter `tileMatrix` with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
	
	`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'`

8.6.4. Parameter tileRow

Requirement 41	/req/tiles/root/tcs-tilerow-definition

A	The operation SHALL support a parameter `tileRow` with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
	
	`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'`

8.6.5. Parameter tileCol

Requirement 42	/req/tiles/root/tcs-tilecol-definition

A	The operation SHALL support a parameter `tileCol` with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
	
	`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`

8.6.6. Parameter Resources

Requirement 43	**/req/tiles/root/tcs-root-definition**

A	The operation SHALL support an optional parameter `resources` with the following characteristics (shown as OpenAPI Specification 3.0 fragment)
	
	`name: resources`
	`in: query`
	`required: false`
	`style: form`
	`explode: false`
	`schema:`
	`  type: array`
	`  items:`
	`  type: string`

B	The parameter `resources` SHALL contain a comma-separated list of geospatial resource identifiers (collectionId's) or a comma-separated list of full URLs to geospatial resource identifiers.
C	Only the geospatial resource identifiers that advertise a link with type=`tiles` in the geospatial resource description SHALL be included.
D	Only geospatial resources that support the same TileMatrixSetId parameter value SHALL be included.

Recommendation 8	**/rec/tiles/root/tcs-root-definition**
A	If the parameter `resources` is missing, and when it is possible and sensible, all geospatial resources supporting the TileMatrixSetId parameter value SHOULD be represented in the tiles.

Permission 2	**/per/tiles/root/tcs-root-definition**

A	If the parameter `resources` is missing and if it is not possible and sensible to represent all geospatial resources 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 geospatial resources.

8.6.7. 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 44	**/req/tiles/root/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.

8.6.8. 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 `resources` contains a resource id of URI that does not exist on the API, the status code of the response is 404.

If the value of the parameter `resources` has a wrong format or combines one of more geospatial resources that 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