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: Charles Heazel

OGC API Tiles

Copyright notice

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

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. 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.

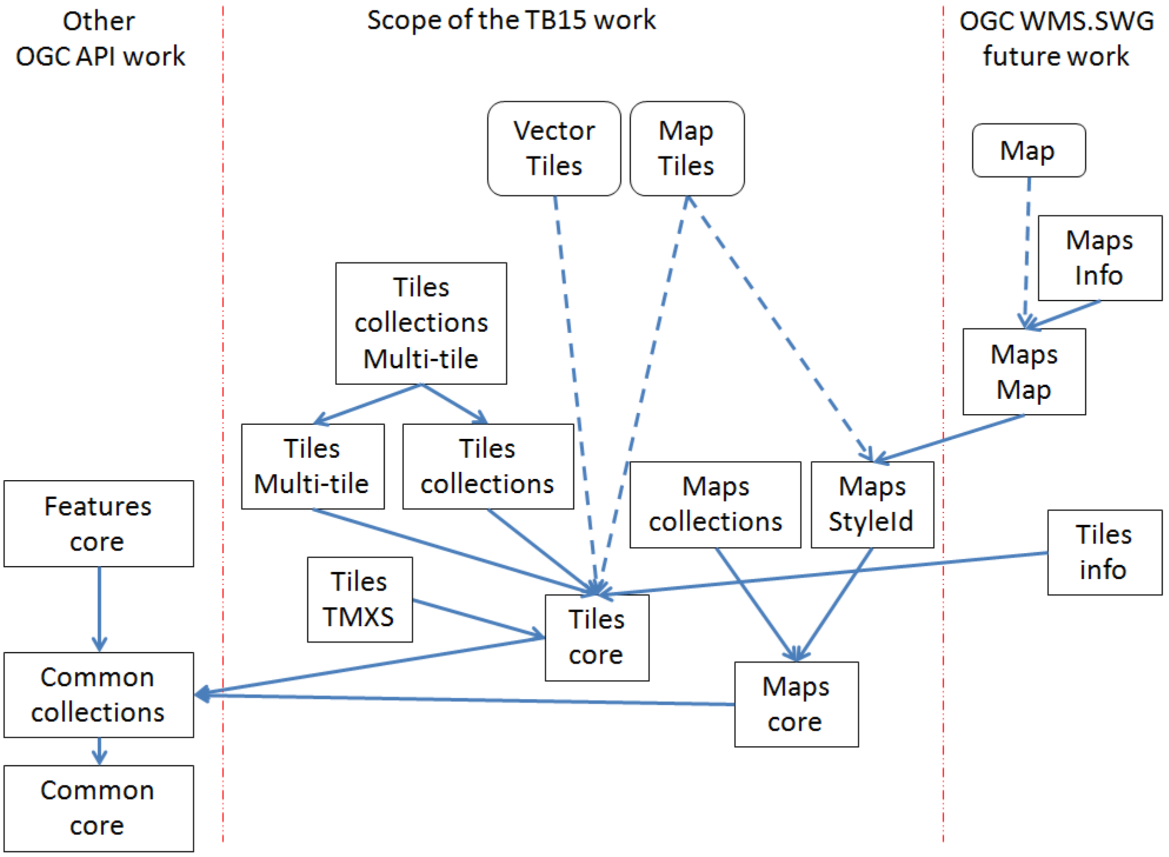

Figure 1. Modular approach in the Maps and Tiles draft specification

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 and 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.

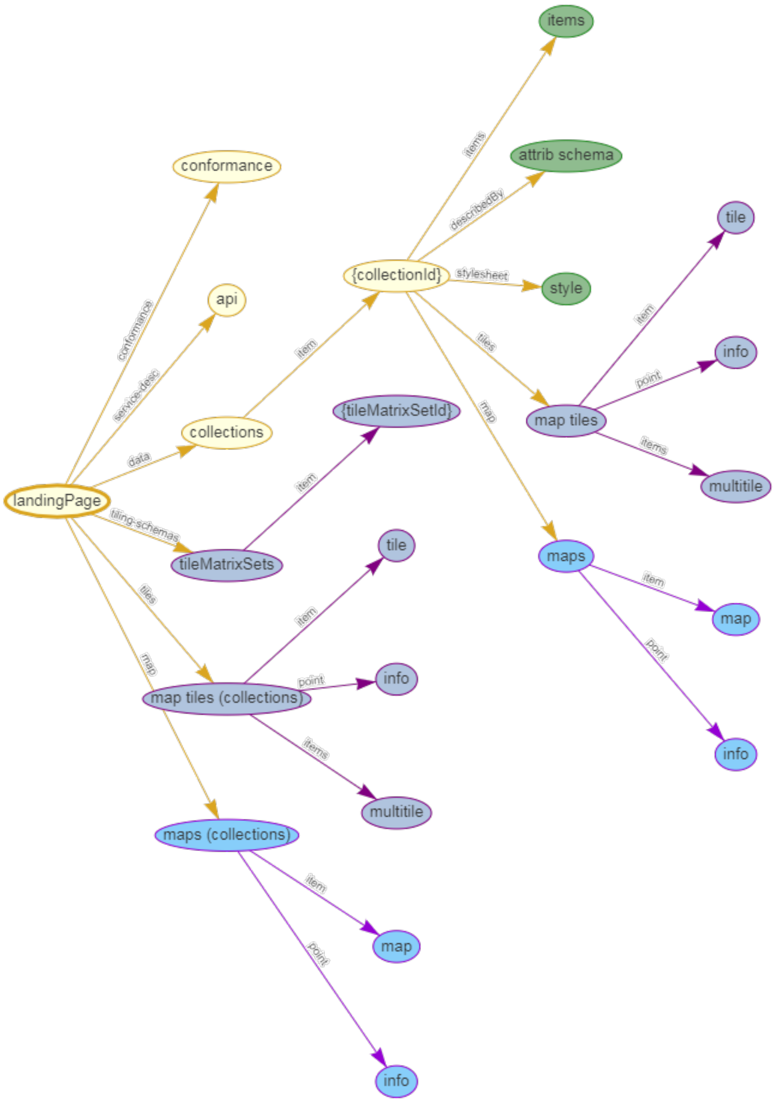

Figure 2. Resources and relations to them via links

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 50	**/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 51	**/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.

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"
  ]
}
```

7.5. 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 52	**/req/tiles/core/tc-md-collection-links**
A	For each collection included in the response, a `links` property of the collection SHALL include a link to the description of the collection (rel: `item`) (in addition to other links specified in OGC API Commons).

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 53	**/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 54	**/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 55	**/req/tiles/core/sct-tmxslink**

<table>
<tr>
<td>A</td>
<td>The content of the response to a successful execution SHALL contain a property called <em>tileMatrixSetLinks</em> with a list of <em>tileMatrixSetLink</em> objects following a data model defined in the clause 7.3 of OGC 17-083r2. In the core specification <em>tileMatrixSetLink</em> 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):
<pre>
  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'
</pre>
</td>
</tr>
</table>

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",
     "rel": "item",
     "type": "image/png",
   }
   ...
  ]
}
```

Recommendation 56	**/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 57	**/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

Example 4. link to get tiles as a URL template in the tiles metadata response fragment

```
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 58	**/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 59	**/req/tiles/core/tc-tilematrixsetid-definition**

<table>
<tr>
<td>A</td>
<td>

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
```

</td>
</tr>
</table>

7.8.3. Parameter tileMatrix

<table>
<tr>
<th>Requirement 60</th>
<th>/req/tiles/core/tc-tilematrix-definition</th>
</tr>
<tr>
<td>A</td>
<td>

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'
```

</td>
</tr>
</table>

7.8.4. Parameter tileRow

Requirement 61	/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'`

7.8.5. Parameter tileCol

Requirement 62	**/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`

7.8.6. 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 63	**/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 8	**/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 64	**/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.

Example 5. 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",
    }
  ]
}
```

8.3. 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 65	**/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*

Example 6. 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/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 66	**/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 67	**/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

Example 7. API tiles response fragment with the link to retrieve tiles form more than one collection

```
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 68	**/req/tiles/collections/tcs-op**
A	The server SHALL support the HTTP GET operation at the path /tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}

8.5.2. Parameter tileMatrixSetId

Requirement 69	**/req/tiles/collections/tcs-tilematrixsetid-definition**

<table>
<tr><td>A</td><td>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
```
</td></tr>
</table>

8.5.3. Parameter tileMatrix

<table>
<tr><th>Requirement 70</th><th>/req/tiles/collections/tcs-tilematrix-definition</th></tr>
<tr><td>A</td><td>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'
```
</td></tr>
</table>

8.5.4. Parameter tileRow

Requirement 71	/req/tiles/collections/tcs-tilerow-definition

<table>
<tr><td>A</td><td>The operation SHALL support a parameter <code>tileRow</code> with the following characteristics (shown as OpenAPI Specification 3.0 fragment):<pre><code> 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'</code></pre></td></tr>
</table>

8.5.5. Parameter tileCol

<table>
<tr><th>Requirement 72</th><th>/req/tiles/collections/tcs-tilecol-definition</th></tr>
<tr><td>A</td><td>The operation SHALL support a parameter <code>tileCol</code> with the following characteristics (shown as OpenAPI Specification 3.0 fragment):<pre><code> 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</code></pre></td></tr>
</table>

8.5.6. Parameter Collections

Requirement 73	/req/tiles/collections/tcs-collections-definition

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

Recommendation 9	**/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 4	**/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 74	**/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.

Chapter 9. Requirement Class "Tiles Tile Matrix Set"

9.1. Overview

Requirements Class	
http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/tmxs	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core
Dependency	http://www.opengis.net/spec/tilematrixset/1.0/req/json-tilematrixset2d
Dependency	http://www.opengis.net/spec/tilematrixset/1.0/req/tilematrixsetlimits2d
Dependency	http://www.opengis.net/spec/tilematrixset/1.0/req/json-tilematrixsetlimits2d
Dependency	http://www.opengis.net/spec/tilematrixset/1.0/req/tilematrixsetlink2d
Dependency	http://www.opengis.net/spec/tilematrixset/1.0/req/json-tilematrixsetlink2d

The *tiles core* requirements class states that the service can support the eight TileMatrixSets defined in the Annex D.1 of the OGC 17-083r2 standard by mentioning their identifiers without the need to describe them. This requirement class acts as and extension of the core requirements class that adds all the necessary elements to support other TileMatrixSets by adding a mechanism to fully describe TileMatrixSets that are specific to the API instance.

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 `TileMatrixSets` (path `/tileMatrixSets`, link relation `tiling-schemes`).

9.2. API landing page

The landing page provides links to start exploring the resources offered by the API. It mainly consists in a list of links. The core of this draft specification does not add anything to the links required by OGC API - Common. This extension for TileMatrixSet requires new links for TileMatrixSets on top of the common ones.

9.2.1. Response

Requirement 75	**/req/tiles/tmxs/root-success**
A	The API SHALL advertise a URI to retrieve the list of TileMatrixSets defined by this service as links to the descriptions paths with rel=`tiling-schemes`.

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

Example 8. API Landing Page fragment with links to TileMatrixSet descriptions

```
{
  links: [
    ...,
    {
      "href": "http://data.example.org/tileMatrixSet?f=json",
      "rel": "tiling-schemas",
      "type": "application/json",
      "title": "List of tileMatrixSets implemented by this API in JSON
",
    },
    {
      "href": "http://data.example.org/tileMatrixSet?f=html",
      "rel": "tiling-schemas",
      "type": "text/html",
      "title": "List of tileMatrixSets implemented by this API in HTML
",
    }
  ]
}
```

9.3. Declaration of conformance classes

9.3.1. Response

The conformance page mainly consists of a list of links. OGC API - Common already requires some links for the core and collections requirements classes

Requirement 76	/req/tiles/tmxs/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/tmxs.

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

Example 9. 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/tmxs"
    "http://www.opengis.net/spec/tilematrixset/1.0/req/tilematrixset2d"
    "http://www.opengis.net/spec/tilematrixset/1.0/req/json-
tilematrixset2d"

"http://www.opengis.net/spec/tilematrixset/1.0/req/tilematrixsetlimits2
d"
    "http://www.opengis.net/spec/tilematrixset/1.0/req/json-
tilematrixsetlimits2d"
  ]
}
```

9.4. TileMatrixSets

The TileMatrixSets operation retrieves links to the descriptions of the tile matrix sets supported by the API instance in addition to the eight TileMatrixSets defined in the Annex D.1 of the OGC 17-083r2 standard.

9.4.1. Operation

Requirement 77	**/req/tiles/tmxs/tmxs-tilematrixsets-op**
A	The server SHALL support the HTTP GET operation at the path /tileMatrixSets.

9.4.2. Response

Requirement 78	**/req/tiles/tmxs/tmxs-tilematrixsets-success**
A	A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.
B	The body of the response SHALL be a tileMatrixSets object listing the tilematrixsets supported by this server other than the eight ones defined in the Annex D of OGC 17-083r2 standard.
C	For each TileMatrixSet the response SHALL contain a TileMatrixSet id and a link to request the TileMatrixSet description.

Example 10. Schema for the TileMatrixSets resource

```
  type: object
  required:
    - tileMatrixSets
  properties:
    tileMatrixSets:
      type: array
      items:
        $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/id-link'
```

Example 11. Schema for id-link from OGC API - Common used in TileMatrixSets resource.

```
id-link:
  type: object
  description: |-
    Reusable object that contains an id to a resource and links where
the object is described or a representation retrieved. Typically it is
useful for paths like `/resources` and `/resources/{resourceId}`.
`/resources` will respond an array of id-link listing the `resourceId`
and the links to get it. /collections and /collections/{collectionId}
is an exception to this pattern.
    The fact that `links` is an array can be used to advertise the same
object representation in different formats.
  required:
    - id
    - links
  properties:
    id:
      type: string
    title:
      type: string
    links:
      type: array
      minItems: 1
      items:
        $ref: '#/components/schemas/link'
```

Example 12. Example for the TileMatrixSets resource

```
{
  "tileMatrixSets": [
    {
      "id": "MyWebMercatorQuad",
      "title": "My Google Maps Compatible for the World",
      "links": [
        {
          "href":
"https://data.example.org/tileMatrixSet/MyWebMercatorQuad",
          "rel": "item",
          "type": "application/json"
        }
      ]
    }
  ]
}
```

9.5. TileMatrixSet

The TileMatrixSet operation retrieves the full description of a tile matrix set supported by the API instance following the schema described in the OGC 17-083r2 standard.

9.5.1. Operation

Requirement 79	**/req/tiles/tmxs/tmxs-tilematrixset-op**
A	The server SHALL support the HTTP GET operation at the path /tileMatrixSet/{tileMatrixSetId}.
A	The parameter tileMatrixSetId is each id property in the tileMatrixSets response.

9.5.2. Response

Requirement 80	**/req/tiles/tmxs/tmxs-tilematrixset-op**

A	A successful execution of the operation SHALL be reported as a response with a HTTP status code 200.
B	The body of the response SHALL follow the TileMatrixSet data model defined in the http://www.opengis.net/spec/tilematrixset/1.0/req/tilematrixset2d requirements class of the Clause 7 in the OGC 17-083r2 standard.
C	The body of the response SHALL be encoded in JSON following the requirements class http://www.opengis.net/spec/tilematrixset/1.0/req/json-tilematrixset2d of the Clause 9 in the OGC 17-083r2 standard.

Recommendation 10	**/rec/tiles/tmxs/tilematrixset-response**
A	The server may support a tileMatrixSetId that is one of the eight TileMatrixSets defined in the Annex D of OGC 17-083r2 and return a successful response with a description identical to the one in the Annex D of OGC 17-083r2.

Example 13. Fragment of a TileMatrixSet resource

```
{
  "title": "Google Maps Compatible for the World",
  "abstract": "The most common TileMatrixSet, used in most of the main
IT map browsers. It was initially popularized by Google Maps",
  "identifier": "WebMercatorQuad",
  "supportedCRS": "http://www.opengis.net/def/crs/EPSG/0/3857",
  "wellKnownScaleSet":
"http://www.opengis.net/def/wkss/OGC/1.0/GoogleMapsCompatible",
  "tileMatrix": [
    ...
    {
      "title": "Google Maps Compatible for the World zoom level 3",
      "abstract": "Google Maps Compatible zoom level 3 that is
equivalent to a scale of 1:69885283.00358972 and has 19567.87924100512
meters of pixel size in the equator",
      "identifier": "3",
      "scaleDenominator": 69885283.00358972,
      "topLeftCorner": [
        -20037508.3427892,
        20037508.3427892
      ],
      "tileWidth": 256,
      "tileHeight": 256,
      "matrixHeight": 8,
      "matrixWidth": 8
    }
    ...
  ]
}
```

9.6. Tiles

The requirements class described in this section also defines and extra element *limits* in the tiles metadata returned by a successful /collection/{collectionId}/tiles request that can be used for the API instance to document limitations in the scales and extents supported in the context of the tile matrix set that is defined in a more unrestricted way.

9.6.1. Collection extra properties

Requirement 81	**/req/tiles/tmxs/stc-limits**

A	If the extent of the available tiles in the server is smaller than the extent of the TileMatrixSet, the object *tileMatrixSetLinks* in the response to a successful execution of the *tiles* request SHALL contain a property called *tileMatrixSetLimits* that is an array that specifies the limitations in the area available for this collection for each TileMatrix. *tileMatrixSetLink* object follows a data model defined in the clause 7.3 of OGC 17-083r2 that can be encoded in the following schema (shown as an OpenAPI Specification 3.0 fragment):

```
  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'
      tileMatrixSetLimits:
        type: array
        minItems: 1
        items:
          $ref:
'#/components/schemas/tileMatrixSetLimits-entry'
  tileMatrixSetLimits-entry:
    type: object
    required:
      - tileMatrix
      - minTileRow
      - maxTileRow
      - minTileCol
      - maxTileCol
    properties:
      tileMatrix:
        type: string
        format: uri
        example: '5'
      minTileRow:
        type: number
        format: integer
```

B	The server SHALL only successfully respond with tiles for the mentioned scales and in the range of tilecol and tilerow defined. If the range of tilecol and tilerow is missing for a scale, all tilecol and tilerow values SHALL be make available by the server for this scale.

```
          example: 1
        minTileCol:
          type: number
```

Example 14. Fragment of a Tiles resource with limits

```
{
  "tileMatrixSetLinks": [
    {
      "type": "tileMatrixSetLink",
      "tileMatrixSet":
"http://www.opengis.net/def/tilematrixset/OGC/1.0/WebMercatorQuad",
      "tileMatrixSetLimits": [
        {
          "type": "tileMatrixSetLimits",
          "tileMatrix": "5",
          "minTileRow": 0,
          "maxTileRow": 1,
          "minTileCol": 3,
          "maxTileCol": 4
        }
      ]
    }
  ],
  ...
  "links": [
    ...
    {
     "href":
"http://data.example.com/collections/buildings/tiles/{tileMatrixSetId}/
{tileMatrix}/{tileRow}/{tileCol}.png",
     "rel": "item",
     "type": "image/png",
     "$$ref": "https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/examples/link-tiles-tile"
   }
   ...
  ]
}
```

Chapter 10. Requirement Class "Tiles Info"

10.1. Overview

NOTE This section should be elaborated by a SWG and only some hints are provided in this Engineering Report

WARNING Some subsections are intentionally left blank.

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

This requirements calls makes data contained in tiles a little more informative than just "nice pictures" by allowing clients to implement a click user event. By clicking on a pixel in the screen that shows a tile, the user will receive some textual information describing what is shown in that pixel. For example, by clicking on a tile containing elevation data the user will get the elevation value.

NOTE The use of `pixel in the screen` can create the wrong impression that this operation is restricted to "raster based tiles". This is not necessarily true. The Two Dimensional Tile Matrix Set standard (OGC 17-083r2) discusses how tile matrices are created for an optimum resolution in the screen, even if they might be entirely feature based.

When fully completed, the new OGC API architecture should be able to integrate several representations of the same resource. This way a digital elevation model could be accessible as a tile and also as a coverage. The coverage part should be able to provide elevation values to the client. When that day arrives, this info requirements class will no longer be needed as the coverage functionality will provide the client with enough data to emulate this extension and some other extra interactions such us the capability to create vertical profiles.

10.2. Overview

TBD

10.3. Declaration of conformance classes

10.3.1. Response

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

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

Example 15. 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/info"
  ]
}
```

10.4. Collection

This draft specification includes dependencies on OGC API - Common collection. The response to the operation is extended with the necessary information to formulate a tile response for this collection.

10.4.1. Collection Links

Example 16. API collection response fragment

```
links:
[
    {
      "href":
"http://data.example.com/collections/buildings/tiles/WorldMercatorWGS84
Quad/0/0/0",
      "rel": "tiles",
      "type": "image/png",
    },
    {
      "href":
"http://data.example.com/collections/buildings/tiles/WorldMercatorWGS84
Quad/0/0/0/info",
      "rel": "attributes",
      "type": "text/html",
    }
]
```

10.5. FeatureInfo

Implementations of the OGC API – Maps and Tiles draft specification may support requests for information about the features present at a particular pixel location in the screen on a map tile. Requests for feature information will specify the tile along with a pixel location on that tile. The server will provide information on the features present at or near the location specified by the client request. The server may choose what information to provide about the nearby features.

10.5.1. FeatureInfo document

A FeatureInfo document is the resource representation of a FeatureInfo resource in resource oriented architectural style. The FeatureInfo document SHALL be in the format specified in the request when that format has been advertised in the **ServiceMetadata document** as available for that FeatureInfo resource.

For better interoperability between servers and clients, the Simple Features Profile of the Geography Markup Language (GML) [06-049r1] as a supported document format for FeatureInfo resources is recommended. The Simple Features Profile of GML defines three levels of content in three profiles with different degrees of constraints to the GML flexibility. Support for the most constrained one (level 0) that results in a simpler GML document is strongly recommended. In the context of that profile only simple XML types can be used as thematic properties and cardinality greater than one is not allowed. Servers and clients SHALL specify the MIME type "application/gml+xml; version=3.1" as an InfoFormat value and the GML application schema of the response SHOULD conform to GML Simple Features profile level 0 when that GML profile is used. In most cases, only thematic attributes of the features are intended to be included in a FeatureInfo document but the Simple Features profiles were evidently intended to include the geometric information of the features in the GML objects. However, an application schema can be generated that does not include feature geometry and only describes non-geometric feature attribute types. This can be very useful to avoid unnecessarily requesting long sequences of position values in line or polygon layers.

Also, to allow easy presentation of the data, support for the HTML format (represented by an InfoFormat MIME type of "text/html") is also recommended.

Chapter 11. Requirement Class "Tiles Multi-tiles"

11.1. Overview

This requirement class opens the possibility exchange multiple tiles covering a bounding box and belonging to one or more scales with a single client-server interaction.

WARNING

Currently, this requirement class does not provide any way that clients or servers can limit the size of the multi-tile response. Even with a relatively small bounding box, the result of a multi-tile request (in particular to a collection that has tiles available at very small scale denominator values) could result in list of tiles too big for the server to generate or for the client to handle. The current specified default values for bounding box and scales range parameters will most probably incur in this problem. Before this requirement class is endorsed by the OGC, this issue should be addressed. One possible solution is to allow the server for specifying a maximum size limit (in kilobytes, or in number of tiles) and to force the server to start from the higher level of scale denominator and stop when the limit is reached. Adding a paging mechanism in the request could help on fragmenting big responses in smaller chunks that can be sequentially requested.

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

In this requirements class, a mechanism to request more than one tile from a single collection in a single request. This mechanism is called a ‘multi-tile’ is defined. The result can be a document listing the needed tiles to cover a bounding box or a package with all tiles inside.

11.2. Declaration of conformance classes

11.2.1. Response

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

Requirement 82	**/req/tiles/multitiles/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/multitiles.

In the conformance page (typically in JSON format) the links follow the link schema defined in the OGC API - Common. The following is an example fragment of the response to an OGC API - Tiles conformance information page with support for multi-tiles.

Example 17. 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/multitiles"
  ]
}
```

11.3. Tiles description

The response to a tiles description request contains the necessary information to later formulate a tile or a multi-tile request for a collection.

11.3.1. 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. This extension adds the URL template to a multi-tile.

Requirement 83	**/req/tiles/multitiles/mtc-multitiles-examples**
A	The content of the response to a successful execution SHALL include at least a link to a multi-tiles URI template (rel: `items`).

B	These links SHALL provide a URL template with the fragment `/tiles` followed by the variables {tileMatrixSetId}. Once the variables are substituted by their respective valid values, a URL to a multitiles is obtained.
C	There SHALL be a link to a multitile 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} 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 specification.

Table 4. URI template variables for tiles and possible values

Example 18. API tiles response fragment

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

11.4. Multiple tiles from one collection

The following requirements provide a mechanism to select and retrieve a set of tiles at once following a TileMatrixSet.

11.4.1. Operation

Requirement 84	**/req/tiles/multitiles/mtc-op**
A	Tiles SHALL be available as HTTP GET requests to a URI that will be composed by two parts: a initial part is the URI of a resource that can be represented as tiles and the final part follows the pattern /tiles/{tileMatrixSetId}
B	Only the resources or collections that advertise one of more links with type=tiles SHALL be requested as multiple tiles.

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

11.4.2. Parameter tileMatrixSetId

<table>
<tr><th>Requirement 85</th><th>/req/tiles/multitiles/mtc-tilematrixsetid-definition</th></tr>
<tr><td>A</td><td>The operation SHALL support a parameter <code>tileMatrixSetId</code> with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
<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>
</td></tr>
</table>

11.4.3. Parameter bbox

Requirement 86	**/req/tiles/multitiles/mtc-bbox-definition**

A	The operation SHALL support an optional parameter bbox to filter the area where tiles will be retrieved with the following characteristics (shown as OpenAPI Specification 3.0 fragment):

```
    name: bbox
    in: query
    description:
      'Only elements that have a geometry that
intersects the bounding box are selected.
      The bounding box is provided as four or six
numbers, depending on whether the coordinate
reference system includes a vertical axis
(elevation or depth):

      * Lower left corner, coordinate axis 1
      * Lower left corner, coordinate axis 2
      * Lower left corner, coordinate axis 3
(optional)
      * Upper right corner, coordinate axis 1
      * Upper right corner, coordinate axis 2
      * Upper right corner, coordinate axis 3
(optional)

      The coordinate reference system of the
values is WGS 84 longitude/latitude

(http://www.opengis.net/def/crs/OGC/1.3/CRS84)
unless a different coordinate reference system is
specified by another parameter in the API (e.g
`bbox-crs`).'
    required: false
    schema:
      type: array
      minItems: 4
      maxItems: 6
      items:
        type: number
        format: double
    style: form
    explode: false
```

B	A TileMatrixSet definition points to a CRS. The coordinates of the bbox SHALL be in the CRS as specified in the definition of the TileMatrixSet identified by the tileMatrixSetId
C	If the ´bbox´ parameter is not specified, the server SHALL assume the whole extent of the tiles is requested.

This definition in inherited from OGC API - Common.

11.4.4. Parameter scaleDenominator

Requirement 87	**/req/tiles/multitiles/mtc-scaledenominator-definition**
A	The operation SHALL support an optional parameter `scaleDenominator` to filter the scales where tiles will be retrieved with the following characteristics (shown as OpenAPI Specification 3.0 fragment):

```
  name: scaleDenominator
   in: query
   description:
      A range of scale denominators (that can be
used to generate a list of tileMatrix names).
   required: false
   style: form
   explode: false
   schema:
     type: array
     minItems: 2
     maxItems: 2
     items:
       type: number
       format: double
```

B	If the parameter is not specified, the server SHALL assume all TileMatrices (scales) SHALL be returned.

Recommendation 11	**/rec/tiles/multitiles/mtc-scaledenominator-definition**

A	To prevent mistakes identifying the scale denominator due to precision issues caused by lack of significant digits, the client should apply a tolerance to intervals. If the client wants to specify a single scale denominator, it will use a small interval with enough tolerance.

11.4.5. Parameter multiTileType

<table>
<tr><th>Requirement 88</th><th>/req/tiles/multitiles/mtc-multitiletype-definition</th></tr>
<tr><td>A</td><td>The operation SHALL support an optional parameter <code>multitileType</code> that determines the type of the response and with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
<pre><code> name: multiTileType
  in: query
  description:
    'When successful, the service will respond to
a query in one of two ways. It can provide a file
with links to each tile or or it will provide the
tiles in a package. The package can still contain
the description of each tile

    The allowed values for this parameter are
`url`, `tiles` and `full`.'
  style: form
  schema:
    type: string
    default: tiles
    enum:
      - url
      - tiles
      - full
  example: full
</code></pre></td></tr>
<tr><td>B</td><td>If the value of the <code>multiTileType</code> parameter is set to <code>url</code>, the server SHALL return a list of the selected tiles in a format following the <code>tileSet</code> schema. Each tile description in the list will contain a URL to download the tile later.</td></tr>
</table>

C	If the value of the `multiTileType` parameter is set to `tiles` or if the parameter is not specified in the request, the server SHALL return a package (e.g. a ZIP file) that will include tiles as separated parts in the package.
D	If the value of the `multiTileType` parameter is set to `full`, the server SHALL return the tiles and a list of the selected tiles (in a format following the `tileSet` schema) as part of a package.

Permission 5	**/per/tiles/multitiles/mtc-multitiletype-definition**
A	The server MAY only implement a subset of the enumerated values (url, tiles, full) for the parameter `multitileType` and in this case it will only enumerate this subset in its schema.

11.4.6. Formats

In the cases of the multi-tile response, there are two formats involved. The multi-tile itself can be returned as a package (e.g. a ZIP file) that contains the tiles inside. The individual tiles also have their format. The format of the multi-tile is governed by the format procedure specified in the OGC API - Common. When the server supports multiple encoding for the individual tiles and the client has a preference for the tiles format, there is a need for communicating this preference to the server. This document does not mandate any particular approach how this is supported but provides the following recommendation.

Recommendation 12	**/rec/tiles/multitiles/mtc-f-tile-definition**
A	When the web interaction allows for HTTP format negotiation `Accept:` header is preferable to specify the required formats. In the case of multi-tile a composed format is recommended following the pattern application/vnd.ogc.multipart;container={multitile-media-type};tiles={tile-media-type}} (example: application/vnd.ogc.multipart;container=application/x-zip-compressed;tiles=image/png)

B	When the web interaction does not allow for controlling the HTTP format negotiation (e.g. URL in a HTML link), the operation MAY support an optional parameter `f-tile` to specify the tile media type that the client prefers and a parameter `f` for the media type of the multi-tile response.
C	The content of these parameters should be specified by the server instance as an enumeration of supported media types in the API description.

11.4.7. Response

A successful response to a set of tiles will be consistent with the media type of resource requested.

Requirement 89	**/req/tiles/multitiles/mtc-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 be inside or intersect with the spatial extent of the geographical area represented by the 'bbox' and `scaleDenominator`.

C

If a list of the tiles has been requested, the content of that response SHALL contain a tileSet document be based upon the following OpenAPI 3.0 schema:

```
  tileSet:
    description: This is the response for a
multiple tiles request.
    type: object
    required: tileSet
    properties:
      tileSet:
        type: array
        items:
          $ref:
'#/components/schemas/tileSetEntry'
  tileSetEntry:
    description:
      This is an entry on a multiple tiles
request.
    type: object
    required:
      - tileURL
      - tileMatrix
      - tileRow
      - tileCol
    properties:
      tileURL:
        type: string
        format: uri
      tileMatrix:
        type: string
      tileRow:
        type: number
      tileCol:
        type: number
      width:
        type: number
        description:
          The width of the tile in rendering
device pixels. If it exceeds the visual display
area be should cut when displayed
      height:
        type: number
        description:
          The height of the tile in rendering
device pixels. If it exceeds the visual display
```

D	When a package is being returned and the package format supports expressing file paths of its parts (such as the ZIP file), each tile in the package SHALL have a path following the template: `{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}.{file-extension}`. {file-extension} is the file extension that corresponds to the media type (e.g "jpg" for image/jpeg).

```
cut when displayed
        left:
          type: number
          description:
            Horizontal position from the left of
the visual display area in pixels. Negative value
means that the left side of the tile is outside
the top-left corner of the display and should be
cut when displayed
```

Mainly this extension suggests 3 possible alternatives for a multi-tile response being the last one (`full`) the combination of the first two (`url` and `package`).

List Response

This format assumes that the client has a viewport to represent an geographic area defined by the bounding box and the scale (that defines the pixel size of the viewport) in the screen. This area should be populated with tiles. The server is expected to enumerate the tiles needed to populate the viewport and optionally to provide information on how to position the tiles in the viewport.

In the following example, we assume that the bounding box and scale provided implies a viewport of 336x446 pixels (height by width). The viewport it covered by 4 tiles. The client has requested a `url` type of multi-tile and negotiated a response a JSON format. The URL of each tile is provided, accompanied with information on the position of the top left corner of each one in the viewport.

Example 19. Example of a tileSet document

```
{
  "tileSet": [
    {
      "tileURL":
"http://data.example.com/collections/buildings/tiles/WebMercatorQuad/2/
0/0.png",
      "tileMatrix": 0,
      "tileRow": 0,
      "tileCol": 0,
      "width": 256,
      "height": 256,
      "top": -10,
      "left": -20
    },
    {
      "tileURL":
"http://data.example.com/collections/buildings/tiles/WebMercatorQuad/2/
0/1.png",
      "tileMatrix": 0,
      "tileRow": 0,
      "tileCol": 1,
      "width": 100,
      "height": 256,
      "top": -10,
      "left": 236
    },
    {
      "tileURL":
"http://data.example.com/collections/buildings/tiles/WebMercatorQuad/2/
1/0.png",
      "tileMatrix": 0,
      "tileRow": 1,
      "tileCol": 0,
      "width": 256,
      "height": 200,
      "top": 246,
      "left": -20
    },
    {
      "tileURL":
"http://data.example.com/collections/buildings/tiles/WebMercatorQuad/2/
1/1.png",
      "tileMatrix": 0,
      "tileRow": 1,
      "tileCol": 1,
      "width": 100,
      "height": 200,
      "top": 246,
      "left": 236
    }
  ]
}
```

Package Response

This format assumes that the client is interested in the tiles that cover a geographic area defined by the bounding box and the scale (or scales). The client knows what to do with the tiles and it is able to identify the tiles by their path using the URI template of the server as a pattern to extract the TileMatrix, TileRow and TileCol of each one.

Assuming that the client has requested a scale that fits with TileMatrix "2" and a bounding box that requires 2x2 tiles and that he client has requested a `package` type of multi-tile and negotiated a ZIP format, a ZIP file is produced and sent by the server with the following files and paths:

File	Path	TileMatrix	TileRow	TileCol
0.png	WebMercatorQuad/2/0	2	0	0
1.png	WebMercatorQuad/2/0	2	0	1
0.png	WebMercatorQuad/2/1	2	1	0
1.png	WebMercatorQuad/2/1	2	1	1

Table 5. Content of a package containing 4 tiles

11.4.8. Error conditions

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

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

Chapter 12. Requirement Class "Tiles Collections Multi-tiles"

12.1. Overview

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

This requirements class defines a mechanism to request more than one tile from more than one collection in a single request. The result can be a document listing the needed tiles to cover a bounding box or a package with all tiles inside. This section shares most of the content with the previous one and intents to provide similar mechanism. The main difference is the capability to request tiles that include elements of multiple collections provided by the parameter 'collections'.

12.2. 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 requirement class of this draft specification does not add anything to the links required by OGC API - Common. The collections extension requires new link for the description of the tiles from more than one collection on top of the common ones that is inherited and needed by this extension.

12.3. Declaration of conformance classes

12.3.1. Response

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

Requirement 90	**/req/tiles/cols-multitiles/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/cols-multitiles.

In the conformance page (typically in JSON format) the links follow the link schema defined in the OGC API - Common. The following is an example fragment of the response of an OGC API - Tiles conformance information page with links to the `collections` requirements class and this requirements class.

Example 20. 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/collections"
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/cols-
multitiles"
  ]
}
```

12.4. Tiles description

The response to this operation contains the necessary information to later formulate a tile request from more than one collection as described in the `collections` extension. This requirement class adds an extra link for the multi-tiles

12.4.1. Response

A successful response to a tiles request for more than on collection will respond with a data structure with specific information necessary to get tiles representing the resource collection. In this extension, the response informs about the URL template to retrieve multi-tiles.

Requirement 91	**/req/tiles/cols-multitiles/mtcs-multitiles-examples**
A	The content of the response to a successful execution SHALL include at least a link to a multi-tiles from multiple collections URI template (rel: `items`).
B	These links SHALL provide a URL template with the fragment `/tiles` followed by the variables {tileMatrixSetId}. Once the variables are substituted by their respective valid values, a URL to a multi-tiles endpoint is obtained.

C	There SHALL be a link to a multi-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}, but 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.

Table 6. URI template variables for tiles and possible values

Example 21. API tiles response fragment

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

12.5. Multiple tiles from more than one collection

This extension provides a mechanism to select and retrieve a set of tiles at once from a TileMatrixSet.

12.5.1. Operation

Requirement 92	/req/tiles/multitiles/mtcs-op
A	Tiles SHALL be available as HTTP GET requests 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}

B	Only the resources or collections that advertise one of more links with type=tiles SHALL be requested as multiple tiles.

12.5.2. Parameter tileMatrixSetId

<table>
<tr><th>Requirement 93</th><th>/req/tiles/multitiles/mtcs-tilematrixsetid-definition</th></tr>
<tr><td>A</td><td>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
```
</td></tr>
</table>

12.5.3. Parameter bbox

Requirement 94	/req/tiles/multitiles/mtcs-bbox-definition

A	The operation SHALL support an optional parameter bbox to filter the area where tiles will be retrieved with the following characteristics (shown as OpenAPI Specification 3.0 fragment):

```
    name: bbox
    in: query
    description:
      'Only elements that have a geometry that
intersects the bounding box are selected.
      The bounding box is provided as four or six
numbers, depending on whether the coordinate
reference system includes a vertical axis
(elevation or depth):

      * Lower left corner, coordinate axis 1
      * Lower left corner, coordinate axis 2
      * Lower left corner, coordinate axis 3
(optional)
      * Upper right corner, coordinate axis 1
      * Upper right corner, coordinate axis 2
      * Upper right corner, coordinate axis 3
(optional)

      The coordinate reference system of the
values is WGS 84 longitude/latitude

(http://www.opengis.net/def/crs/OGC/1.3/CRS84)
unless a different coordinate reference system is
specified by another parameter in the API (e.g
`bbox-crs`).'
    required: false
    schema:
      type: array
      minItems: 4
      maxItems: 6
      items:
        type: number
        format: double
    style: form
    explode: false
```

B	A TileMatrixSet definition points to a CRS. The coordinates of the bbox SHALL be in the CRS of specified in the definition of the TileMatrixSet identified by the tileMatrixSetId
C	If the parameter is not specified, the server SHALL assume the whole extent of the tiles are requested.

This definition in inherited from OGC API - Common.

12.5.4. Parameter scaleDenominator

<table>
<tr><th>Requirement 95</th><th>/req/tiles/multitiles/mtcs-scaledenominator-definition</th></tr>
<tr><td>A</td><td>The operation SHALL support an optional parameter <code>scaleDenominator</code> to filter the scales where tiles will be retrieved with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
<pre>
  name: scaleDenominator
   in: query
   description:
      'A range of scale denominators (that can be
used to generate a list of tileMatrix names).'
   required: false
   style: form
   explode: false
   schema:
     type: array
     minItems: 2
     maxItems: 2
     items:
       type: number
       format: double
</pre></td></tr>
<tr><td>B</td><td>If the parameter is not specified, the server SHALL assume all TileMatrices (scales) SHALL be returned.</td></tr>
</table>

Recommendation 13	/rec/tiles/multitiles/mtcs-scaledenominator-definition

A	To prevent mistakes identifying the scale denominator due to precision issues caused by lack of significant digits, the client should apply a tolerance to intervals. If the client wants to specify a single scale denominator, it will use a small interval with enough tolerance.

12.5.5. Parameter multiTileType

<table>
<tr><th>Requirement 96</th><th>/req/tiles/multitiles/mtcs-multitiletype-definition</th></tr>
<tr><td>A</td><td>The operation SHALL support an optional parameter <code>multitileType</code> that determines the type of the response and with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
<pre><code> name: multiTileType
  in: query
  description:
    'When successful, the service will respond to
 a query in one of two ways. It can provide a file
 with links to each tile or or it will provide the
 tiles in a package. The package can still contain
 the description of each tile

    The allowed values for this parameter are
 `url`, `tiles` and `full`.'
  style: form
  schema:
    type: string
    default: tiles
    enum:
      - url
      - tiles
      - full
  example: full
</code></pre></td></tr>
<tr><td>B</td><td>If the value of the <code>multiTileType</code> parameter is set to <code>url</code> the server SHALL return a list of the selected tiles in a format following the <code>tileSet</code> schema. Each tile description in the list will contain a URL to download the tile later.</td></tr>
</table>

C	If the value of the multiTileType parameter is set to tiles or if the parameter is not specified in the request, the server SHALL return a package (e.g. a ZIP file) that will include tiles as separated parts in the package.
D	If the value of the multiTileType parameter is set to full the server SHALL return the tiles and a list of the selected tiles (in a format following the tileSet schema) as part of a package.

Permission 6	**/per/tiles/multitiles/mtcs-multitiletype-definition**
A	The server MAY only implement a subset of the enumerated values (url, tiles, full) for the parameter multitileType and in this case the server will only enumerate this subset in its schema.

12.5.6. Parameter Collections

<table>
<tr><th>Requirement 97</th><th>/req/tiles/collections/mtcs-collections-definition</th></tr>
<tr><td>A</td><td>The operation SHALL support an optional parameter collections with the following characteristics (shown as OpenAPI Specification 3.0 fragment)
<pre><code>name: collections
in: query
required: false
style: form
explode: false
schema:
  type: array
  items:
    type: string</code></pre></td></tr>
<tr><td>B</td><td>collections SHALL contain a comma-separated list of collection identifiers.</td></tr>
<tr><td>C</td><td>Only the collections that advertise a link type=tiles in the /collections/{collectionId} SHALL be included.</td></tr>
</table>

D	Only the collections that support the same TileMatrixSetId parameter value SHALL be included
C	If `collections` is missing, all collections supporting the TileMatrixSetId parameter value will be considered.

12.5.7. Formats

In the cases of the multi-tile response, there are two formats involved. The multi-tile itself can be returned as a package (e.g. a ZIP file) that contains the tiles inside. The individual tile also has its own format. The format of the multi-tile is governed by the format procedure specified in the OGC API – Common draft specification. When the server supports multiple encodings for the individual tiles and the client has a preference for the tiles format, there is a need for communicating this preference to the server. This document does not mandate any particular approach for how this is supported but provides the following recommendation.

Recommendation 14	**/rec/tiles/multitiles/mtcs-f-tile-definition**
A	The operation MAY support an optional parameter `f-tile` to specify the tile format that the client prefers as parts of the multi-tile response.
B	The content of this parameter should be specified by the server instance as an enumeration of supported media types.

12.5.8. Response

A successful response for a set of tiles will be consistent with the media type of the resource requested. This draft specification does not impose any media type but suggests the use of a package format.

Requirement 98	**/req/tiles/cols-multitiles/mtcs-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 be inside or intersect with the spatial extent of the geographical area represented by the 'bbox' and `scaleDenominator`.

C	If a list of the tiles has been requested, the content of that response SHALL contain a tileSet document be based upon the following OpenAPI 3.0 schema:

```
 tileSet:
  description: This is the response for a
multiple tiles request.
  type: object
  required: tileSet
  properties:
   tileSet:
    type: array
    items:
     $ref:
'#/components/schemas/tileSetEntry'
 tileSetEntry:
  description:
   This is an entry on a multiple tiles
request.
  type: object
  required:
   - tileURL
   - tileMatrix
   - tileRow
   - tileCol
  properties:
   tileURL:
    type: string
    format: uri
   tileMatrix:
    type: string
   tileRow:
    type: number
   tileCol:
    type: number
   width:
    type: number
    description:
     The width of the tile in rendering
device pixels. If it exceeds the visual display
area be should cut when displayed
   height:
    type: number
    description:
     The height of the tile in rendering
device pixels. If it exceeds the visual display
```

D	When a package is being returned and the package format supports expressing file paths of its parts (such as the ZIP file), each tile in the package SHALL have a path following the template: `{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}.{file-extension}`. {file-extension} is the file extension that corresponds to the media type (e.g "jpg" for image/jpeg).

```
cut when displayed
    left:
      type: number
      description:
          Horizontal position from the left of
the visual display area in pixels. Negative value
means that the left side of the tile is outside
the top-left corner of the display and should be
cut when displayed
```

12.5.9. Error conditions

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

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

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