OGC API-Tiles

OGC API Tiles

Warning

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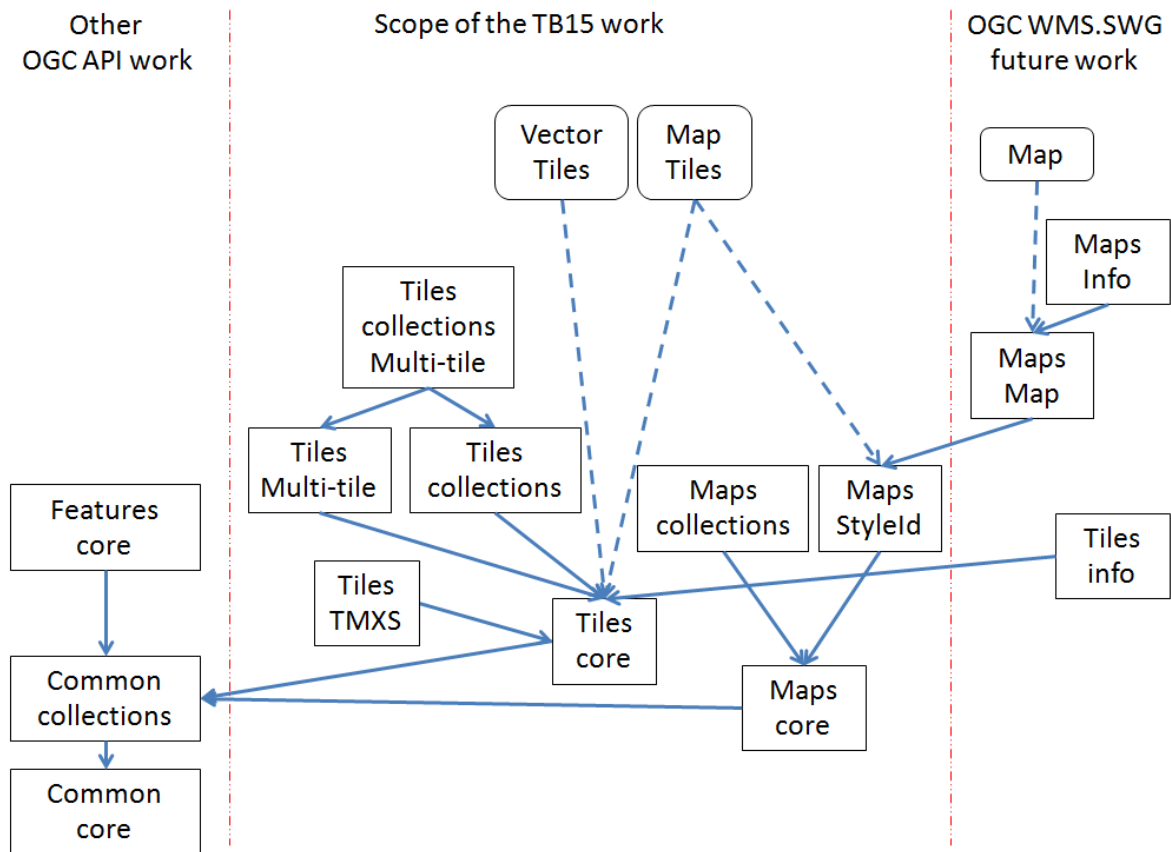
Figure 1. Modular approach in the Maps and Tiles draft specification

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.
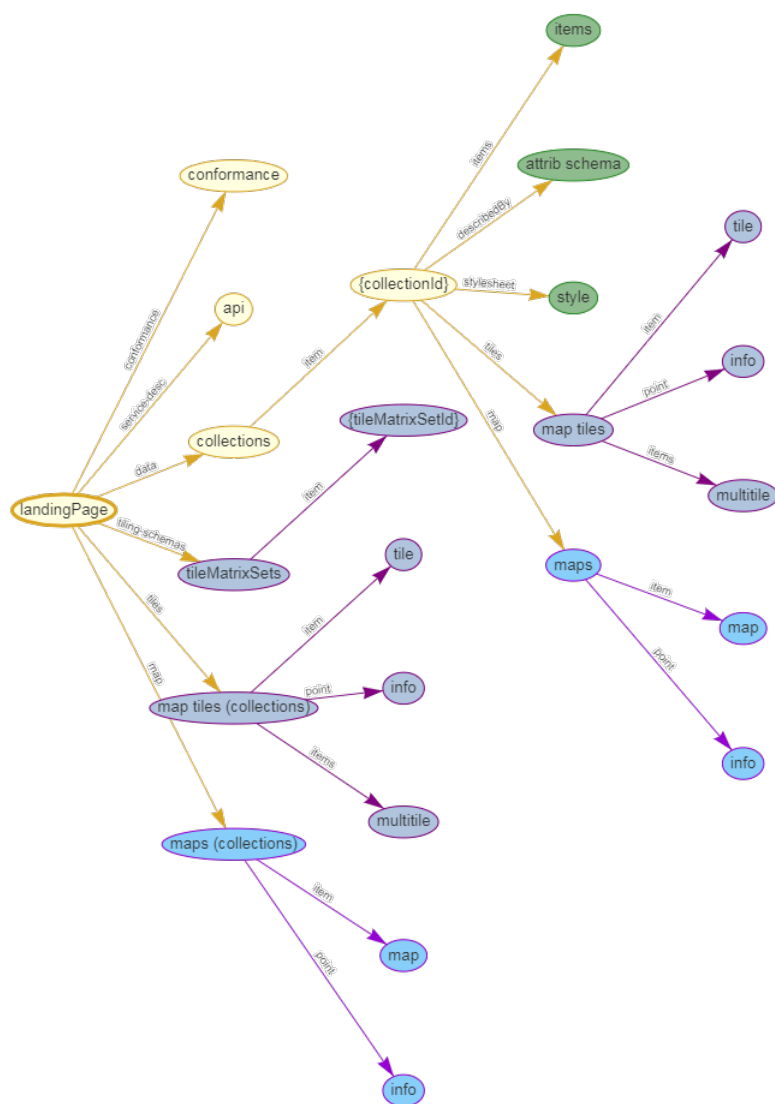
Figure 2. Resources and relations to them via links

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 Tile Matrix Set"

7.1. 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`).

7.2. 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.

7.2.1. Response

Requirement 8	/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 1. 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",
    }
  ]
}
```

7.3. Declaration of conformance classes

7.3.1. 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 9	/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 2. 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"
  ]
}
```

7.4. 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.

7.4.1. Operation

Requirement 10	/req/tiles/tmxs/tmxs-tilematrixsets-op
A	The server SHALL support the HTTP GET operation at the path /tileMatrixSets.

7.4.2. Response

Requirement 11	/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 3. 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 4. Schema for id-link from OGC API - Common used in TileMatrixSets resource.

```yaml
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'
```

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

7.5. 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.

7.5.1. Operation

Requirement 12	/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.

7.5.2. Response

Requirement 13	/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 2	/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 6. 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
    }
    ...
  ]
}
```

7.6. 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.

7.6.1. Collection extra properties

Requirement 14	/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
            minimun: 0
            example: 0
          maxTileRow:
```

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.

```
format: integer
minimum: 0
example: 3
```

Example 7. Fragment of a Tiles resource with limits

```json
{
  "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"
    }
    ...
  ]
}
```

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