OGC API-Tiles

Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: 2019-03-06

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

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

Version: 0.0.1

Category: OGC® Implementation Specification

Editor: Joan Masó

OGC API Tiles

Warning

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

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

Document type: OGC®ImplementationSpecification
Document subtype: if applicable
Document stage: Draft
Document language: English

License Agreement

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

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

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

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

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

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

Table of Contents

i. Abstract

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

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

ii. Keywords

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

ogcdoc, OGC document, tiling, WMTS

iii. Preface

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

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

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

iv. Submitting organizations

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

Organization name(s)

v. Submitters

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

Name Affiliation

Chapter 1. Scope

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

1.1. Current scope:

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

Chapter 2. Conformance

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

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

The main requirements class is:

- Core.

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

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

C**apture additional requirements classes here**

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

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

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

Chapter 3. References

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

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

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

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

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

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

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

Chapter 4. Terms and Definitions

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

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

4.1. 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.

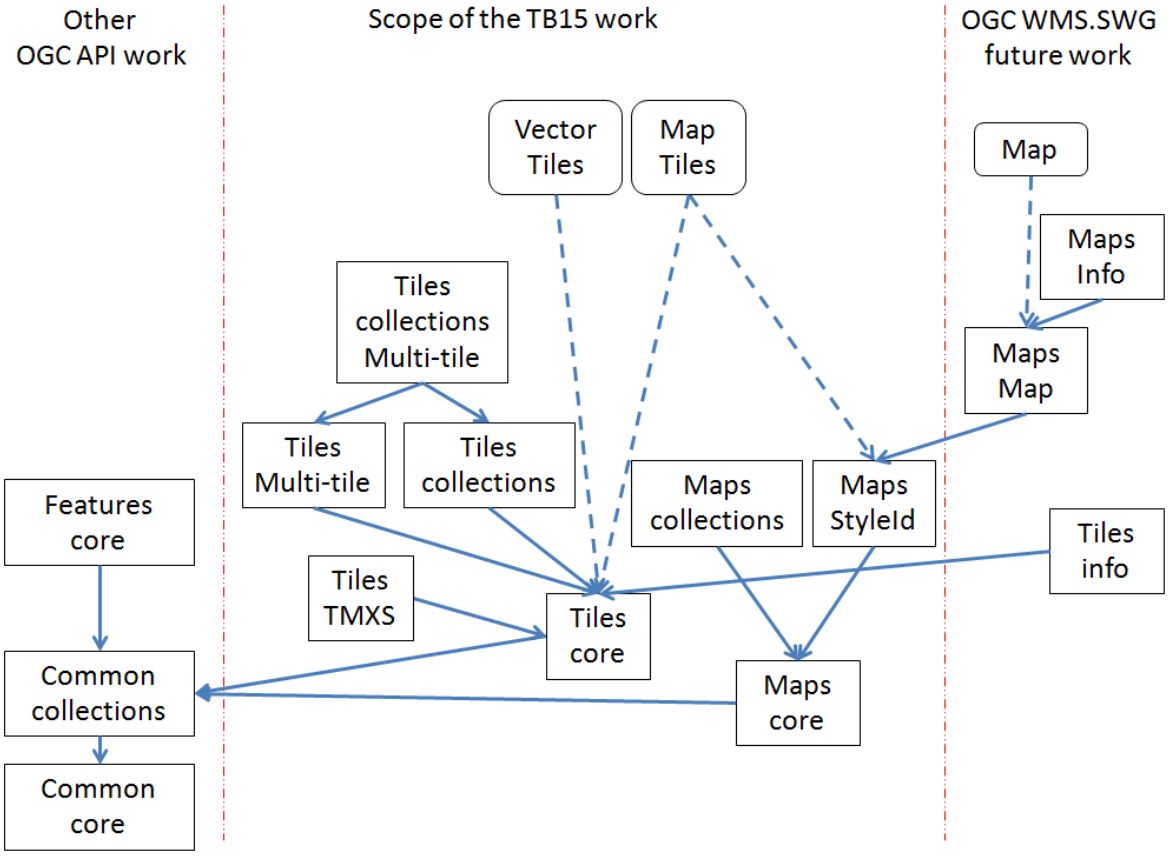

Figure 1. Modular approach in the Maps and Tiles draft specification

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.

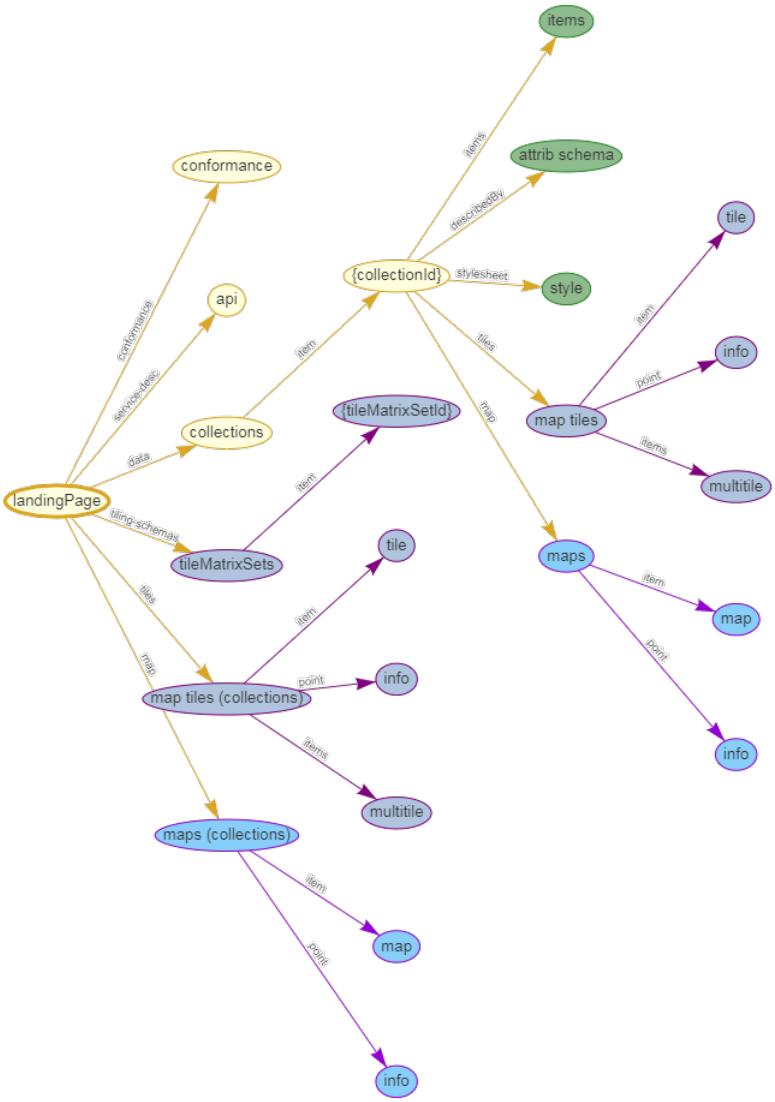

Figure 2. Resources and relations to them via links

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 Multi-tiles"

7.1. 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.

7.2. Declaration of conformance classes

7.2.1. Response

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

Requirement 18	**/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 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"
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/multitiles"
  ]
}
```

7.3. 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.

7.3.1. 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 19	**/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 2. URI template variables for tiles and possible values

Example 2. API tiles response fragment

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

7.4. Multiple tiles from one collection

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

7.4.1. Operation

Requirement 20	/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}).

7.4.2. Parameter tileMatrixSetId

Requirement 21	/req/tiles/multitiles/mtc-tilematrixsetid-definition

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

7.4.3. Parameter bbox

Requirement 22	/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.

7.4.4. Parameter scaleDenominator

Requirement 23	**/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 5	**/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.

7.4.5. Parameter multiTileType

Requirement 24	**/req/tiles/multitiles/mtc-multitiletype-definition**

A	The operation SHALL support an optional parameter `multitileType` that determines the type of the response and with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
	
	`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`

B	If the value of the `multiTileType` parameter is set to `url`, the server SHALL return a list of the selected tiles in a format following the `tileSet` schema. Each tile description in the list will contain a URL to download the tile later.

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 3	**/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.

7.4.6. 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 6	**/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.

7.4.7. Response

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

Requirement 25	**/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
area be should cut when displayed
      top:
        type: number
```

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).

```
        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 3. 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 3. Content of a package containing 4 tiles

7.4.8. 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 8. Requirement Class "Tiles Collections Multi-tiles"

8.1. 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'.

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 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.

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 26	/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 4. 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"
  ]
}
```

8.4. 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

8.4.1. 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 27	**/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 4. URI template variables for tiles and possible values

Example 5. API tiles response fragment

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

8.5. 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.

8.5.1. Operation

Requirement 28	/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.

8.5.2. Parameter tileMatrixSetId

Requirement 29	**/req/tiles/multitiles/mtcs-tilematrixsetid-definition**

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

8.5.3. Parameter bbox

Requirement 30	**/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.

8.5.4. Parameter scaleDenominator

Requirement 31	**/req/tiles/multitiles/mtcs-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 7	**/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.

8.5.5. Parameter multiTileType

Requirement 32	**/req/tiles/multitiles/mtcs-multitiletype-definition**

A	The operation SHALL support an optional parameter `multitileType` that determines the type of the response and with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
	
	`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`

B	If the value of the `multiTileType` parameter is set to `url` the server SHALL return a list of the selected tiles in a format following the `tileSet` schema. Each tile description in the list will contain a URL to download the tile later.

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 4	**/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.

8.5.6. Parameter Collections

Requirement 33	**/req/tiles/collections/mtcs-collections-definition**

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

B	`collections` SHALL contain a comma-separated list of collection identifiers.
C	Only the collections that advertise a link type=tiles in the /collections/{collectionId} SHALL be included.

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

8.5.7. 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 8	**/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.

8.5.8. 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 34	**/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 area be should cut when displayed`
	`  top:`
	`  type: number`

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).

```
        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
```

8.5.9. 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