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 Application Programming Interface (API) for tiled data and map tiles. The API described in this standard builds on the Web Map Tile Service (WMTS) OGC standard. WMTS provides 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. Whereas WMTS focused on map tiles, the OGC API - Tiles standard has been designed to support any form of tiled data. For example, the API can support map tiles, vector tiles, tiled gridded coverages and other tiled data.

ii. Keywords

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

ogcdoc, OGC document, tiling, WMTS, map tiles, vector tiles, tiled feature data

iii. Preface

This document defines the OGC API - Tiles standard. An API conforming to this standard can serve tiles of spatially referenced data or maps 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
Joan Masó	UAB-CREAF
Chuck Heazel	Heazel Tech
Jeff Harrison	AGC
Jérôme Jacovella-St-Louis	Ecere Corporation
TBA	TBA
TBA	TBA

Chapter 1. Scope

This document specifies the behavior of Web APIs that provide access to tiles of one geospatial data resource or more than one geopatial data resource. This standard defines how to discover which resources offered by the API can be retrieved as tiles, what are the tile matrix sets supported by the geospatial data resources, which are the limits of the tiled space and how to request one tile at a time.

A web API can be combine this standard with other standards to extend the scope of the API by adding functionality.

Chapter 2. Conformance

This standard defines two requirements / conformance classes.

The standardization targets of all conformance classes are "Web APIs."

The main requirements class is:

- Core.

The Core specifies requirements that all Web APIs have to implement is they are claiming to support tiles from one geopatial resource following the OGC API - Tiles Part 1: core standard.

Another requirements class is:

- Root

The Root specifies requirements that all Web APIs have to implement is they are claiming to support tiles from more than one geopatial resource following the OGC API - Tiles Part 1: core standard.

This document does not mandate a specific encoding or format for representing tiles. There are no requirements classes making considerations on specify representations.

Core and *Root* are comformance classes that act as building blocks that should be implemented in combination with other more fundamental conformance classes that provide support for API discovery, conformity and API formal definition (e.g. OpenAPI). Possible altermantives for these fundamental conformance classes are OGC API - Common Part 1: core, OGC API - Features Part 1: core or any other non-OGC classes that provide this functionality.

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.

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

NOTE	Commonly, this document is expected to be used in combination with OGC: OGC 19-072, OGC API - Common Part 1: core, but this is not a requirement so it is not listed as a normative reference. For example, OGC API - Features Part 1: core could be used instead.

Chapter 4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of OGC Web Services Common [https://portal.opengeospatial.org/files/?artifact_id=38867] (OGC 06-121r9), 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.

This document also uses terms defined in Sub-clause "Terms and Definitions" of OGC API - Common, Part 1: core.

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

4.1. geospatial data resource

web resource that consists in a set of geospatial data

NOTE	OGC API collections are considered geospatial data resource by definition

4.2. geospatial aspect

web resource that represents a component of geospatial information (metadata, schemas,…) or geospatial data in a particular data model (e.g. feature items, tiles, coverages,…) of a more generic geospatial data resource (e.g. a collection)

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

4.3. map tile

tile that contains information in a gridded form. Commonly the values of the grid represent colors of each cell in the grid for immediate pictorial representation on rendering devices, but can also be coverage subsets.

NOTE	From OGC 17-083r2: OGC Two Dimensional Tile Matrix Set
NOTE	Map tiles are generated in combination with the OGC API - Maps. This tiles are sometimes refered as *raster tiles*

4.4. tile

geometric shape with known properties that may or may not be the result of a tiling (tessellation) process. A tile consists of a single connected "piece" without "holes" or "lines" (topological disc).

NOTE	From OGC 19-014r1: Core Tiling Conceptual and Logical Models for 2D Euclidean Space

small rectangular representation of geographic data, often part of a set of such elements, covering a tiling scheme and sharing similar information content and graphical styling. A tile can be uniquely defined in a tile matrix by one integer index in each dimension. Tiles are mainly used for fast transfer (particularly in the web) and easy display at the resolution of a rendering device. Tiles can be grid based pictorial representations, coverage subsets, or feature based representations (e.g., vector tiles).

NOTE	From OGC 17-083r2: OGC Two Dimensional Tile Matrix Set

4.5. tile matrix

grid tile set scheme defining how space is partitioned into a set of regular conterminous tiles at a fixed scale giving tile scheme to each tile.

NOTE	From OGC 17-083r2: OGC Two Dimensional Tile Matrix Set (modified).

NOTE	For the purposes of this document a tile matrix is restricted to a tessellation of the space that resembles a matrix in a 2D space characterized by a matrix width (columns) and a matrix height (rows).

4.6. tile matrix set

tile set scheme composed by collection of tile matrices defined at different scales covering approximately the same area and having a common coordinate reference system.

NOTE	From OGC 17-083r2: OGC Two Dimensional Tile Matrix Set but modified.

4.7. tile scheme

scheme that defines the unique properties of each individual tile in a tile set. These properties include a unique identifier for each tile, the tile origin, and the extent of the tile.

4.8. tile set

set of tiles - a collection of subsets of the space being partitioned.

NOTE	From OGC 19-014r1: Core Tiling Conceptual and Logical Models for 2D Euclidean Space

NOTE	For the purposes of this document, a tile set is a series of actual tiles contain data and following a common tile set scheme.

4.9. tile set scheme

scheme that defines how space is partitioned into individual tiled units. A tiling scheme defines the spatial reference system, the geometric properties of a tile, which space a uniquely identified tile occupies, and reversely, which unique identifier corresponds to a space satisfying the geometric properties to be a tile.

NOTE	From OGC 19-014r1: Core Tiling Conceptual and Logical Models for 2D Euclidean Space

NOTE	A tile set scheme ca be defined on top of a CRS as well as other spatial reference systems such as DGGS and other organizations including irregular ones. In this document, only tile set schemes based on CRSs are supported.

4.10. vector tile

tile that contains vector information that has been simplified at the tile scale resolution and clipped by the tile boundaries.

NOTE	From OGC 17-083r2: OGC Two Dimensional Tile Matrix Set but modified.

4.11. Web API

API using an architectural style that is founded on the technologies of the Web [source: OGC API - Features - Part 1: Core]

NOTE	See Best Practice 24: Use Web Standards as the foundation of APIs [https://www.w3.org/TR/dwbp/#APIHttpVerbs] (W3C Data on the Web Best Practices) for more detail.

Chapter 5. Conventions

This section provides details of conventions used in this document.

5.1. Identifiers

The normative provisions in this standard are denoted by the URI `http://www.opengis.net/spec/ogcapi-tiles-1/1.0`.

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

5.2. Link relations

To express relationships between resources, RFC 8288 (Web Linking) [https://tools.ietf.org/rfc/rfc8288.txt] is used.

The following IANA link relation types [https://www.iana.org/assignments/link-relations/link-relations.xhtml] are used in this document:

- **alternate**: Refers to a substitute for this context.

- **self**: Conveys an identifier for the link's context.

- **item**: The target IRI points to a resource that is a member of the collection represented by the context IRI.

In addition, the following link relation types are used for which no applicable registered link relation type could be identified:

- **tiles**: The target IRI points to a resource that describes how to provide tiles of the context resource.

Used in combination with OGC API - Features Part 1: core or OGC API Common other link relation types will be used, being the most common: * **conformance**: Refers to a resource that identifies the specifications that the link's context conforms to. * **service-desc**: Identifies service description for the context that is primarily intended for consumption by machines. (API definitions are considered service descriptions) * **service-doc**: Identifies service documentation for the context that is primarily intended for human consumption.

5.3. Use of HTTPS

For simplicity, this document in general only refers to the HTTP protocol. This is not meant to exclude the use of HTTPS and simply is a shorthand notation for "HTTP or HTTPS." In fact, most servers are expected to use HTTPS [https://tools.ietf.org/html/rfc2818], not HTTP [https://www.ietf.org/

rfc/rfc2616.txt].

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. The Web Map Tile Service 1.0 already defines a resource oriented architechtural style but it lack an API defintion. This OGC API - Tiles standard 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.

This document provides de necessary elements to incorporate tile support to an API implementation. These elements can be incorporated in an API based on the OGC API - Features Part 1, core or can be incorporated in an API implementation based on the OGC API – Common Part 1, core. Both specifies a kernel of an API approach to services that follows current resource-oriented architecture practices in the OGC. The OGC API - Common standard provides the foundation upon which implementations of the OGC APIs can be built. The OGC API - common can be combined with this standard and other resource-specific OGC API standards to build and OGC API implementation. However, this standard is done in a way that can extend OGC API - Common but does not make OGC API - Common mandatory. This way This standard it can be reused as a building block in other frameworks.

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

- Clear separation between core requirements and more advanced capabilities. This OGC API – Tiles standard presents the requirements that are relevant for almost everyone who wants to share or use Tiled Data on a fine-grained level. Additional capabilities that several communities are using today will be specified as extensions to the Core API.

- Technologies that change more frequently are decoupled and specified in separate modules ("requirements classes" in OGC terminology). This enables, for example, the use/re-use of new encodings for spatial data or API descriptions.

- Modularization is not just about a single "service". OGC APIs will provide building blocks that can be reused in APIs in general. In other words, a server supporting the OGC API - Tiles should not be seen as a standalone service. Rather it should be viewed as a collection of API building blocks which together implement Tile capabilities. A corollary for this is that it should be possible to implement an API that simultaneously conforms to conformance classes from the Feature, Coverage, Map, Tiles, and other future OGC Web API standards.

This approach intends to support two types of client developers:

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

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

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

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

6.2. Tiles and maps

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

With time, the concept of a tile, initially used for *map tiles* has been generalized to other data models such as feature data (some vendors use the expression *vector tiles*) and even to coverage data or processes that can be parallelized dividing the space in tiles. This standard presents an approach to tiles that can be applied to almost every resource type that returns geospatial data. 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 standard and on top of a collection that is transformed into a map by applying a style, the result should be map tiles (usually in PNG or JPEG format).

This document is part of a series of OGC API - Tiles "parts" that use the core and extensions model. This document includes the "core" and the "root" extension. It is foreseen that future parts will specify other extensions such as how to defined tile matrix sets, how to get tiles from more that one geospatial resource, how to get information of a point in a tile and how to get multiple tiles in a single request. The OGC API Maps is required only to the extent to allow for map tiles to be created on top of a map created by selecting a geospatial data resource with style (or multiple geospatial data resource with styles). An additional extension will deal with time, elevation and other dimensions.
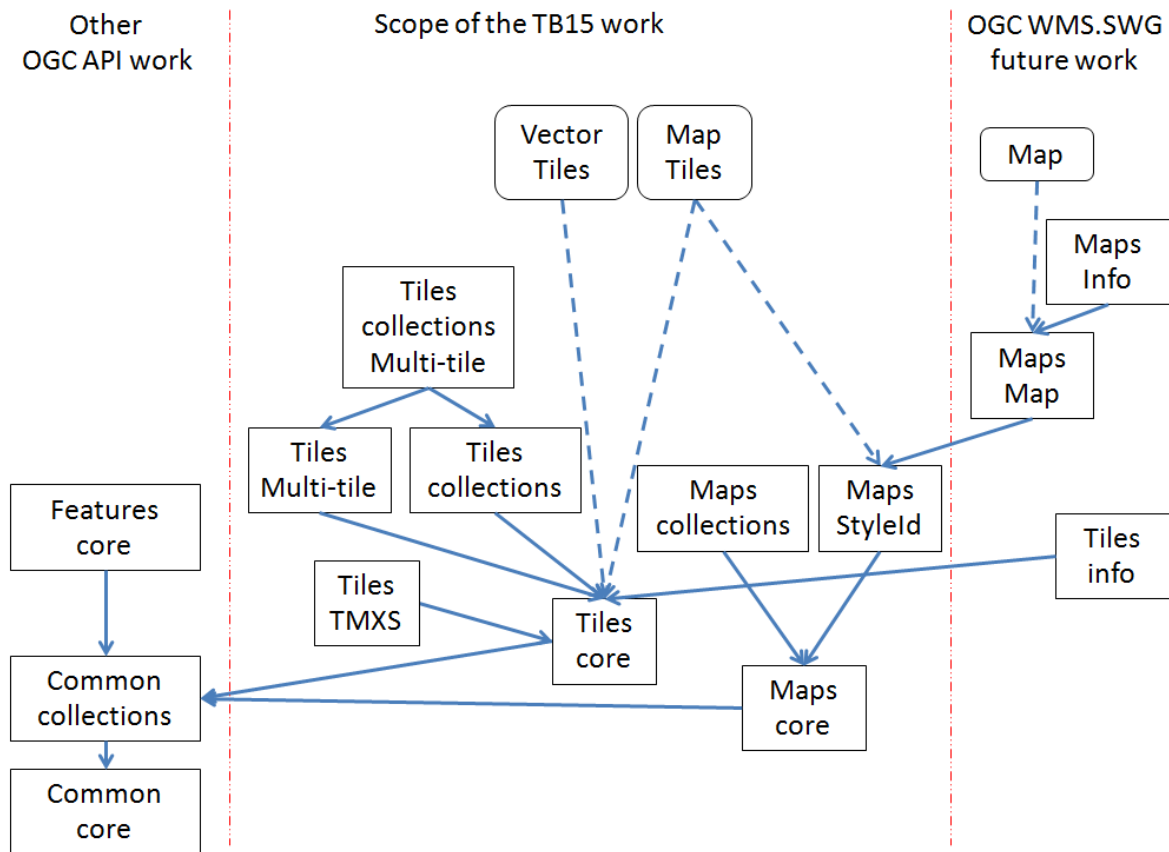
Figure 1. Modular approach in the Maps and Tiles draft specification

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 and path templates 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.
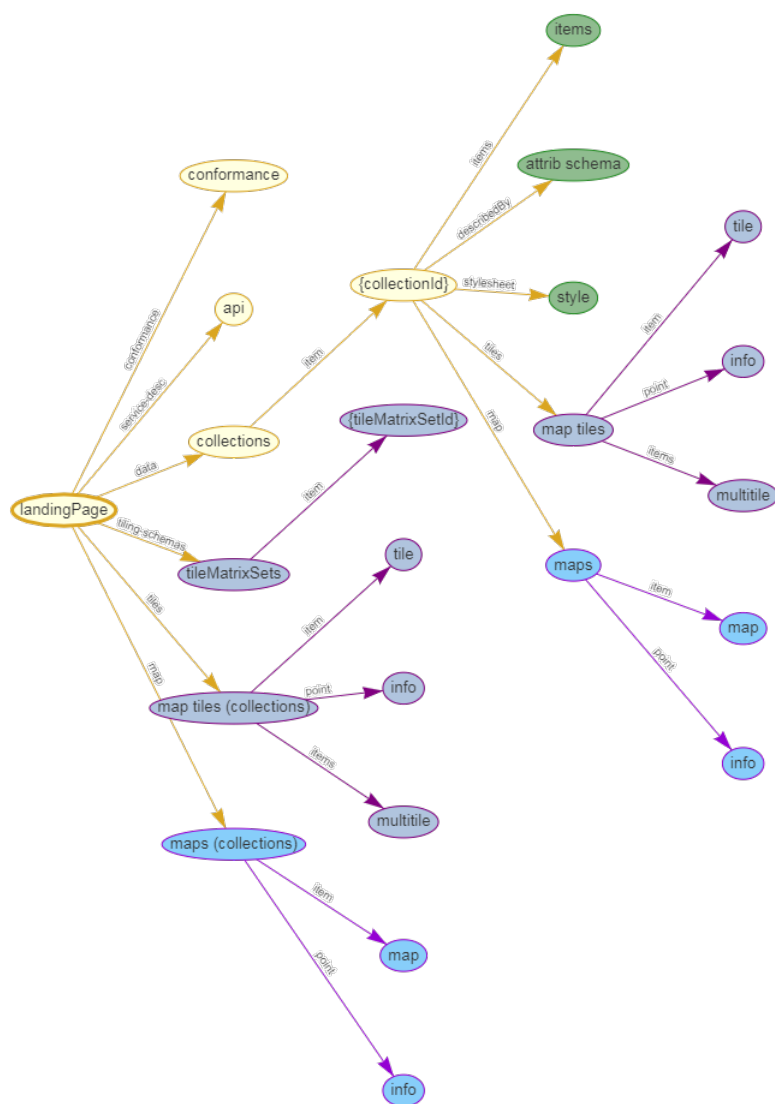
Figure 2. Resources and relations to them via links

For the second approach, we provide some examples of paths templates that an OpenAPI definition documents commonly enumerate. The paths ans path templates can be used to get the necessary resources directly.

Resource name	Common path
Landing page[4]	/
Conformance declaration[4]	/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 in one tile matrix set[2]	`/collections/{collectionId}/tiles/{tileMatrixSetId}`
Vector Tile	`/collections/{collectionId}/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}`
Vector Tiles description (geospatial resources[1])	`/tiles`
Vector Tile	`/collections/{collectionId}/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}`
Vector tile (geospatial resources[1])	`/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}`
Maps	
Maps description[3]	`/collections/{collectionId}/map`
Maps description (geospatial resources[1])[3]	`/map`
Map tiles	
Map tiles description	`/collections/{collectionId}/map/tiles`
Map tiles description in one tile matrix set[2]	`/collections/{collectionId}/map/tiles/{tileMatrixSetId}`
Map tiles description (geospatial resources[1])	`/map/tiles`
Map tiles description (geospatial resources[1]) in one tile matrix set[2]	`/map/tiles/{tileMatrixSetId}`
Map tile	`/collections/{collectionId}/map/{styleId}/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}`
Map tile (geospatial resources[1])	`/map/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}`

Table 1. Overview of resources and common direct links that can be used to define an OGC API and can be described in an Open API document

[1]: The expression "geospatial resources" means "from more than one geospatial resource or collection" [2]: Specified in an annnex of this document, providing support for the "TileJSON" format. [3]: Specified in the OGC API - Maps Part1, core [4]: Specified in the OGC API - Commonn Part1, core

NOTE	Depite the fact that the previous list of path and path templates are used in many implementations of the OGC API - Tiles, the use of these exact paths are not require by this document and others are possible if correctly described in both approaches.

Chapter 7. Requirement Class "TileSet"

7.1. Overview

Requirements Class	
http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/tileset	
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

This is a building block for the an OGC API that describes the TileSet resource and how to retrieve a tile from a tileset

Defines the response of a TileSet GET request (but not the path or the link for the request). The response has two formats: tilejson and OGC json. The OGC json includes tilematrixset links, a central point (optional), a links to the URI template to individual tiles, and the geospatial data resources involved in the creation of the tiles (optional), and other metadata (including attribution; optional) Defines how to formulate a request for individual tiles and how the response will look like (success and failure)

TBD

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

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

The core of the OGC API - Tiles core draft specification does not mandate the inclusion of an explicit definition of any TileMatrixSet. This draft specification assumes that clients and services know about the eight TileMatrixSets defined in OGC 17-083r2 annex D (or compatible future update of it) and there is no need to define new TileMatrixSets. An extension to the core provides the capability to include definitions of flexible TileMatrixSets that are explicitly defined.

This draft specification assumes that data is organized into one or more geospatial data resources (e.g. the "collections" in OGC API - Features - Part 1: Core [http://www.opengis.net/doc/IS/

ogcapi-features-1/1.0]). Geospatial data resources are referenced using URIs.

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

Accessing the geospatial data resource content (other than as tiles) or its descriptions is out of the scope of this draft specification. If a description of the geospatial data resource exists and it has a mechanism to add links to it, this specification will indicate the need to add a link to the tile description.

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

End of TBD

7.2. 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.

7.2.1. Response

The conformance page mainly consists of a list of links.

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

If the server declares also conformity to OGC API - Common or to OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0], then it has to consider the OGC API - Common requirements for declaring conformance, i.e. the use of a the conformance page. In the JSON format the conformance page is an array of links following the link schema defined in the OGC API - Common or in OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0]. Below is an example fragment of a conformance information page of an API conformant to OGC API - Common and OGC API - Tiles.

Example 1. Conformance Information Page fragment

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

7.3. Tileset resource

A tileset resource represents the set of tiles that belongs to a particular tilematrixset. The resource contains the necessary metadata to enable a client application to formulate a tile request.

7.3.1. Tileset path

This class does not specify the path for a tileSet. Generally, TileSet resources are linked from a Tilesets resource. Refer to the TileSets class to know how TileSets can link to TileSet resources. Refer to the DatasetTileSets and GeoDataResourceTileSets classes that describes two mechanisms to associate tileSets to datasets and to geospatial data resources respectively.

7.3.2. Response

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

Requirement 24	/req/tileset/tmxslink
A	If the tiles are available in a tile matrix set different from WebMercatorQuad, the content of the response to a successful execution to a tileset request SHALL provide the necessary information to formulate a tile request.
B	The minimum required information is the name of the tilematrixset and a URI template for reaching the tiles. Other information can be limits in the ranges of the tile rows and columns as well as links to the geospatial data sources

Recommendation 12	/rec/tileset/tmxslink
A	This requirements class does not provide any mechanism to define 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 [http://docs.opengeospatial.org/is/17-083r2/17-083r2.html#61].
| B | Implementation SHOULD a JSON format that does the link to a tilematrixset using a property called *tileMatrixSetLink* with a *tileMatrixSetLink* object following a data model defined in the clause 7.3 of OGC 17-083r2. In the core specification *tileMatrixSetLink* is only used for referencing the supported TileMatrixSets for the tiles, limiting it to the following schema (expressed as an OpenAPI Specification 3.0 fragment):<br><br>```
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'
``` |
| C | 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. |

	D	This standard recommends the use of the TileMatrixSets defined in Annex D [http://docs.opengeospatial.org/is/17-083r2/17-083r2.html#61] of OGC 17-083r2. In the case of variable-width tiles, the standard recommends the use of the TileMatrixSets defined in Annex H [http://docs.opengeospatial.org/is/17-083r2/17-083r2.html#104] of OGC 17-083r2.

Clients or servers are not required to support a specific default TileMatrixSet.

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 25	/req/tiles/core/sct-tile-template
A	The content of the response to a successful execution of a tileset 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 value of the tileMatrixSet identifier and a URI template that makes use of the following URL template variables {tileMatrix}, {tileRow}, {tileCol}. Once the variables are substituted by their respective valid values, a URL to a tile is obtained.
C	There SHALL be a link to a tile URI template for each file format that the server supports (the format is indicated in the 'type' attribute of the link)
D	A property `templated` SHALL be part of the link properties to indicate that the link needs to be processed to substitute the templated variables with valid values before being used as a URL to a tile.
E	The results of this operation SHALL return a single tile that represents information coming from geospatial data resources.

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

Recommendation 13	/rec/tileset/sourcelink
A	In addition to the mandatory link to the tiles template, link to the geospatial data resource that originated the tiles can be provided with the rel=data. This is particularly useful if the tiles resource are composed by more than one geospatial data resource

Example 2. Example of a tileset response

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

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

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

URL template variable	Meaning	Possible values
TileMatrixSetId	tile matrix set identifier	One of the identifiers included in Annex D of OGC 17-083r2 or an identifier defined by extensions of this core
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

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

NOTE	TileJSON is an alternative for a tileset description document. See Annex TBD for more details about this format. Currently this specification only supports WebMercatorQuad TileMatrixSet and the reference to it is implicit. The version 3 of this specification provides a mechanism to cite data sources too.

7.4. A tile

A tile resource is a geospatial resource of a fragment of a more bigger geospatial data resource that is spatially constrained at the boundaries of the selected tile in a tile matrix set.

7.4.1. Tile path and link

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

7.4.2. Operation

This operation allows retrieving a single tile that represents information coming from geospatial data resources.

Requirement 26	/req/tileset/tc-op
A	A tile that contains available data SHALL be available as a HTTP GET request to a URI that will be composed by two parts: The first part is the URI of a geospatial data resource that can be represented as tiles, followed by the word 'tiles', followed by the a URL template that makes use of the following URI template variables {tileMatrixSetId}, {tileMatrix}, {tileRow}, {tileCol} (the variable {tileMatrixSetId} can be omitted if only one tile matrix set available in the server) (the variable {tileMatrix} can also be omitted in the rare case that there is only one tile matrix set and it only defined one tile matrix). Once the variables are substituted by their respective valid values, a URL to a tile is obtained.

Typical geospatial data resources that can be retrieved as tiles are: features (in OGC API - Features - Part 1: Core [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0] represented by /collections/{collectionId}), or full maps (specified in the OGC API Maps).

Recommendation 14	/rec/tileset/tc-op
A	A tiles implementation should consider to use the tiles URI template variables in the following common order and form: {tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}
B	A tiles implementation should consider to specify the variables {tileMatrixSetId} and {tileMatrix} even if there is only one valid value for them.

NOTE	Clients should not assume the common order of URI template variable and should extract them from the examples in the "links" section or from the API description path templates.

The supported encodings, or more precisely the media types of the supported encodings, can be determined from the API definition. The desired encoding is selected using HTTP content negotiation.

7.4.3. Parameter tileMatrixSetId

Requirement 27	/req/tileset/tc-tilematrixsetid-definition

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

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

7.4.4. Parameter tileMatrix

Requirement 28	/req/tileset/tc-tilematrix-definition
A	The operation SHALL support a parameter `tileMatrix` with the following characteristics (shown as OpenAPI Specification 3.0 fragment):
| | ```
name: tileMatrix
in: path
description: Identifier selecting one of the
scales defined in the TileMatrixSet and
representing the scaleDenominator the tile.
required: true
schema:
  type: string
example: '11'
``` |

7.4.5. Parameter tileRow

Requirement 29	/req/tileset/tc-tilerow-definition

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

7.4.6. Parameter tileCol

Requirement 30	/req/tileset/tc-tilecol-definition

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

7.4.7. Response

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

- For features the media type may 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 31	/req/tileset/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.

Permission 4	/per/tileset/tc-core-tile-encoding
A	This draft specification does not impose any media type on the encoding of a response containing tiled feature data. For features the media type MAY be GeoJSON, Mapbox vector tiles or other format.
B	This draft specification does not impose any media type on the encoding of a response containing tiled coverage data. For coverages it MAY be a GeoTIFF or other format.
C	This draft specification does not impose any media type on the encoding of a map tile response. For maps it MAY be a JPEG, PNG or other format.

Normally, the content partially outside the tile bounding box will be clipped at the extent of the bounding box. This can be done efficiently when tiles are in raster format (e.g. map tiles). However, tiles containing features in vector format may not clip features that are partially outside to ensure continuity of features or for performance.

Recommendation 15	/rec/tileset/tc-success-scale

A	The content of that response should be simplified to comply with the scale denominator represented by the TileMatrix identified. Full resolution geographical elements are only expected for the lower values of scale denominators.

To enable search engines to easily discover the content offered by an implementation of OGC API - Tiles, as well as to enable web browsers to easily display the content offered by the APIs, this standard allows for responses to operations to be encoded in HTML.

Permission 5	**/per/tileset/tc-core-html**
A	Every `200`-response of an operation of the server MAY support the media type `text/html`.

7.4.8. Error conditions

A general summary of the HTTP status codes can be found in OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0] as well as in OGC API - Common.

If the parameter value `tileMatrixSetId` is not available by the server for this resource or the parameters values `tileMatrix`, `tileRow`, `tileCol` are out-of-range, or the tile is not provided due to lack of data in the area, the status code of the response will be `404`.

Chapter 8. Requirement Class "TileSets"

8.1. Overview

Requirements Class	
http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/tilesets	
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-tiles-1/1.0/req/tileset

Defines the response of a TileSets GET request that includes a link to the tileset or an array for tileset responses. It defines the paths to the tileset resources.

TBD

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

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

The core of the OGC API - Tiles core draft specification does not mandate the inclusion of an explicit definition of any TileMatrixSet. This draft specification assumes that clients and services know about the eight TileMatrixSets defined in OGC 17-083r2 annex D (or compatible future update of it) and there is no need to define new TileMatrixSets. An extension to the core provides the capability to include definitions of flexible TileMatrixSets that are explicitly defined.

This draft specification assumes that data is organized into one or more geospatial data resources (e.g. the "collections" in OGC API - Features - Part 1: Core [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0]). Geospatial data resources are referenced using URIs.

This document does not specify any requirements for the type of geospatial data resource that should be supported. Provided that the geospatial data resources can be organized into tiles, they can be supported regardless of whether they are features, coverages, a resource that does

not represent data per-se (e.g. an annotation) and so forth. The resource path replaces the concept of layer in WMS and WMTS. In this core tiles can be generated from only one geospatial data resource (tiles that are generated as a combination of geospatial data resources will be defined as an extension).

Accessing the geospatial data resource content (other than as tiles) or its descriptions is out of the scope of this draft specification. If a description of the geospatial data resource exists and it has a mechanism to add links to it, this specification will indicate the need to add a link to the tile description.

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

End of TBD

8.2. 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.

8.2.1. Response

The conformance page mainly consists of a list of links.

Requirement 32	/req/tilesets/conformance-success
A	If the API has a mechanism to advertise conformance classes, the API SHALL advertise the tilesets conformance class with a link to http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/tilesets.

If the server declares also conformity to OGC API - Common or to OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0], then it has to consider the OGC API - Common requirements for declaring conformance, i.e. the use of a the conformance page. In the JSON format the conformance page is an array of links following the link schema defined in the OGC API - Common or in OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0]. Below is an example fragment of a conformance information page of an API conformant to OGC API - Common and OGC API - Tiles.

Example 3. Conformance Information Page fragment

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

8.3. Tilesets

A tilesets resource represents a list of sets of tiles each one belonging to a particular tilematrixset. The resource contains the necessary metadata to enable a client application to formulate a tile request.

8.3.1. Tilesets path

This class does not specify the path for a tileSets. Refer to the DatasetTileSets and GeoDataResourceTileSets classes that describes two mechanisms to associate tileSets to datasets and to geospatial data resources respectively.

8.3.2. Response

A successful GET response to a tiles description resource will respond with a data structure that lists the tileset URLS available (one for each Tile Matrix Set supported).

Requirement 33	/req/tilesets/tileset-links
A	The content of the response to a successful execution of a tile description SHALL include links to all tileset supported by the tilesets (rel: `item`).

Example 4. Example of a tiles metadata response

```
{
  "links": [
    ...
    {
     "href":
"http://data.example.com/collections/buildings/tiles/WebMercatorQuad.pn
g",
      "rel": "item",
      "type": "application/json",
    }
    ...
  ]
}
```

Chapter 9. Requirement Class "Dataset Tile Sets"

9.1. Overview

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

Defines how to get tiles from the dataset (or datasets) represented by the services. it will tell the path to get a tilesets resources

TBD

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

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

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

End of TBD

9.2. General

Recommendation 16	/rec/dataset-tilesets/api-common
A	An implementation this standard should consider to implement the requirements specified in the http://www.opengis.net/spec/ OAPI_Common/1.0/req/core Requirements Class of the OGC API-Common version 1.0 Standard.

This building block stays flexible and does not require implementation OGC API - Common, allowing for other API architectures outside the OGC API framework to adopt it. However, a server

under the OGC APIs is expected to implement OGC API - Common. If so, in practice, this means that the landing page and the conformance page follow OGC API - Common core and collections requirement classes when used. Temporarily, it is also possible to combine this building block with OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0] that is not tied to OGC API - Common.

9.3. API landing page

The landing page provides links to start exploring the resources offered by the API. It mainly consists of a list of links to root resources. This standard extension requires a new link in the landing page for getting a description of the URL that allows for retrieving tiles of the resources in the dataset

9.3.1. Response

Requirement 34	/req/dataset-tilesets/landingpage
A	If the API has mechanism to expose root resources (e.g. a landing page), the API SHALL advertise URI to retrieve every tileset supported by this service as links to the descriptions paths with rel: `tiles`.
B	The tileset SHALL be available as a URI that will be composed by two parts: The first part is the word '/tiles/', followed by the id of the tile matrix set supported.

In the landing page, in JSON format, the links follow the link schema defined in the OGC API - Common or in OGC API - Features v1. Below you can find an example fragment of the response to an OGC API - Tiles landing page showing the link to dataset tilesets.

Example 5. API Landing Page fragment that advertises dataset tilesets

```
{
  links: [
    ...,
    {
      "href": "http://data.example.org/tiles/WebMercatorQuad",
      "rel": "tiles",
      "type": "application/json",
      "title": "Link to information on map tileset from the dataset",
    }
  ]
}
```

Recommendation 17	/rec/dataset-tilesets/landingpage
A	If the API has mechanism to expose root resources (e.g. a landing page), the API can advertise a URI to retrieve the tilesets defined by this service as links to the descriptions paths with rel: tilesets.
B	The URI will be '/tiles'

9.4. Declaration of conformance classes

9.4.1. Response

The conformance page mainly consists of a list of links.

Requirement 35	/req/dataset-tilesets/conformance-success
A	If the API has a mechanism to advertise conformance classes, the API SHALL advertise the capability of generating tiles from the dataset adding the conformance class for this capability as a link to http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/dataset-tilesets.

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 dataset tilesets

Example 6. Conformance Information Page fragment

```
{
  "conformsTo": [
    "http://www.opengis.net/spec/ogcapi-common-1/1.0/conf/core",
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/tileset"
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/dataset-
tilesets"
  ]
}
```

9.5. tilesets

The response to the tileset operation contains the necessary information to later formulate a tile request of tiles from the dataset.

9.5.1. Operation

Requirement 36	/req/dataset-tilesets/operation
A	The server SHALL support an operation to retrieve tileset's from the dataset as a HTTP GET request to a URI that is composed by three 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/` followed by the identified of the tile matrix set.

The request of this operation has no parameters.

9.5.2. Response

A successful response to a tileset request is a data structure defined by the tileset class

9.6. Tiles

The Tileset class have information on how to retrieve a single tile representing a piece of information in the dataset in the way explained in the tileset class

9.6.1. Response

Recommendation 18	/rec/dataset-tilesets/geodata-selection
A	When it is possible and sensible, all geospatial data resources supporting the TileMatrixSetId parameter value SHOULD be represented in the tiles.

Permission 6	/per/dataset-tilesets/geodata-selection
A	If it is not possible and sensible to represent all geospatial data resources in tiles (e.g. it compromises performance or tiles are become packed with too many elements), the server is allowed to select only the most significant geospatial data resources.

Chapter 10. Requirement Class "Geo Data Resource Selection"

10.1. Overview

Requirements Class	
http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/geodata-selection	
Target type	Web API
Dependency	http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/tileset
Dependency	http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/tilesets
Dependency	http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/dataset-tilesets

It will define how to include a query parameter to subsets of the dataset (e.g. which geospatial data resources are involved in the creation tiles.

TBD

End of TBD

10.2. General

Recommendation 19	/rec/geodata-selection/api-common
A	An implementation this standard should consider to implement the requirements specified in the http://www.opengis.net/spec/OAPI_Common/1.0/req/core and http://www.opengis.net/spec/OAPI_Common/1.0/req/collections Requirements Classes of the OGC API-Common version 1.0 Standard.

This building block stays flexible and does not require implementation OGC API - Common, allowing for other API architectures outside the OGC API framework to adopt it. However, a server under the OGC APIs is expected to implement OGC API - Common. If so, in practice, this means that the landing page and the conformance page follow OGC API - Common core and collections requirement classes when used. Temporarily, it is also possible to combine this building block with OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0] that is not tied to OGC API - Common.

10.3. Declaration of conformance classes

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

10.3.1. Response

The conformance page mainly consists of a list of links.

Requirement 37	/req/geodata-selection/conformance-success
A	If the API has a mechanism to advertise conformance classes, the API SHALL advertise the geospatial data resource selection conformance class with a link to http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/geodata-selection.

If the server declares also conformity to OGC API - Common or to OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0], then it has to consider the OGC API - Common requirements for declaring conformance, i.e. the use of a the conformance page. In the JSON format the conformance page is an array of links following the link schema defined in the OGC API - Common or in OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0]. Below is an example fragment of a conformance information page of an API conformant to OGC API - Common and OGC API - Tiles.

Example 7. Conformance Information Page fragment

```
{
  "conformsTo": [
    "http://www.opengis.net/spec/ogcapi-common-1/1.0/conf/core",
    "http://www.opengis.net/spec/ogcapi-common-1/1.0/conf/collections",
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/tileset"
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/tilesets"
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/dataset-
tilesets"
    "http://www.opengis.net/spec/ogcapi-tiles-
1/1.0/conf/tileset/geodata-selection"
  ]
}
```

10.4. Tiles

The Tilesets have information on how to retrieve a single tile representing a piece of information in the dataset in the way explained in the tileset class

10.4.1. Operation

The operation for retrieving dataset-tilesets is defined in the geodata-selection class. By default, the geospatial data resources included in the dataset-tiles is undefined but it is supposed to represent the dataset. This class adds a mechanism to select the geospatial data resources that are going to be used to create the tiles. In practice this give use the capability to create tiles from more than one geospatial data resources selectively.

10.4.2. Parameter Resources

Requirement 38	/req/geodata-selection/query-geodata

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

B	The parameter geodata SHALL contain a comma-separated list of geospatial resource identifiers (collectionId's) or a comma-separated list of full URLs to geospatial resource identifiers.
C	Only the geospatial data resource identifiers that advertise a link compatible with the type of resource to be requested (with type=tiles for tiles and with type=map for maps) in the geospatial data resource description SHALL be included.

D	Only geospatial data resources that support the same crsID or the sames TileMatrixSetId parameter value SHALL be combined in the parameter value.	
E	If `geodata` is missing, all geospatial data resources supporting the `crsId` or the `TileMatrixSetId` parameter value SHALL be considered.	

10.4.3. Error conditions

If the value of the parameter `geodata` contains a resource id of URI that does not exist on the API, the status code of the response is 404.

If the value of the parameter `geodata` has a wrong format or combines one of more geospatial data resources that are not compatible with the `tileMatrixSetId` value, the status code of the response is 500.

Chapter 11. Requirement Class "Geo Data Resource TileSets"

11.1. Overview

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

It will define how to specify the a link to tilesets containing a representation of this geospatial data resource (path).

TBD

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

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

The core of the OGC API - Tiles core draft specification does not mandate the inclusion of an explicit definition of any TileMatrixSet. This draft specification assumes that clients and services know about the eight TileMatrixSets defined in OGC 17-083r2 annex D (or compatible future update of it) and there is no need to define new TileMatrixSets. An extension to the core provides the capability to include definitions of flexible TileMatrixSets that are explicitly defined.

This draft specification assumes that data is organized into one or more geospatial data resources (e.g. the "collections" in OGC API - Features - Part 1: Core [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0]). Geospatial data resources are referenced using URIs.

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

Accessing the geospatial data resource content (other than as tiles) or its descriptions is out of

the scope of this draft specification. If a description of the geospatial data resource exists and it has a mechanism to add links to it, this specification will indicate the need to add a link to the tile description.

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

End of TBD

11.2. General

Recommendation 20	/rec/geodata-tilesets/api-common
A	An implementation this standard should consider to implement the requirements specified in the http://www.opengis.net/spec/OAPI_Common/1.0/req/core and http://www.opengis.net/spec/OAPI_Common/1.0/req/collections Requirements Classes of the OGC API-Common version 1.0 Standard.

This building block stays flexible and does not require implementation OGC API - Common, allowing for other API architectures outside the OGC API framework to adopt it. However, a server under the OGC APIs is expected to implement OGC API - Common. If so, in practice, this means that the landing page and the conformance page follow OGC API - Common core and collections requirement classes when used. Temporarily, it is also possible to combine this building block with OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0] that is not tied to OGC API - Common.

11.3. Declaration of conformance classes

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

11.3.1. Response

The conformance page mainly consists of a list of links.

Requirement 39	/req/geodata-tileset/conformance-success

A	If the API has a mechanism to advertise conformance classes, the API SHALL advertise the geospatial data resource tilesets conformance class with a link to http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/geodata-tileset.

If the server declares also conformity to OGC API - Common or to OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0], then it has to consider the OGC API - Common requirements for declaring conformance, i.e. the use of a the conformance page. In the JSON format the conformance page is an array of links following the link schema defined in the OGC API - Common or in OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0]. Below is an example fragment of a conformance information page of an API conformant to OGC API - Common and OGC API - Tiles.

Example 8. Conformance Information Page fragment

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

11.4. Geospatial data resources

This standard does not specify how geospatial data resources are exposed in the API and if they can be retrieved as geospatial data (e.g. feature items). For example OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0] includes the definition of collections and each collection is available in the /collections/{collectionId} path. OGC API - Common will provide a similar mechanism. Other paths in the API could also give access to geospatial data resources.

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

Requirement 40	**/req/geodata-tilesets/desc-links**

A	If the API has a mechanism for their geospatial data resources to expose links to geospatial aspects (e.g. feature items, metadata…), the API SHALL include a `link` with the `href` pointing to every tileset supported by the geospatial data resource that presents a tile aspect of this geospatial data resource and with `rel: "tiles"`.
A	The tileset SHALL be available as a URI that will be composed by three parts: The first part is the URL of the geospatial data resources followed by the word '/tiles/', and finally followed by the id of the tile matrix set supported.

For example, an implementation of the OGC API - Features - Part 1: Core [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0] returns a list of links that include geospatial aspects for each geospatial data resource in the /collections/{collectionId} path. OGC API - Common is expected to provide a similar mechanism. In the JSON response, the array `links` is the place for adding a resource reference to the 'tiles' description.

Example 9. Fragment of a collection with a links array with one item of the array pointing to a tilesets.

```
"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/WebMercatorQuad",
    "rel": "tiles",
    "type": "application/json",
  }
]
}
```

Recommendation 21	/rec/geodata-tilesets/desc-links

A	If the API has a mechanism for their geospatial data resources to expose links to geospatial aspects (e.g. feature items, metadata…), the API SHALL include a `link` with the `href` pointing to a tilesets that presents a tile aspect of this geospatial data resource and with `rel: "tilesets"`.
B	The URI will be composed be the path of the geospatial data resources followed by '/tiles'

11.5. Geospatial data resources Tileset

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

11.5.1. Tilesets path

Requirement 41	/req/geodata-tilesets/operation
A	Every geospatial data resource available as tiles SHALL support an path URL and a HTTP GET operation to retrieve the description of the tiles the API implementation can provide
B	The URI shall be composed by two parts: the initial part is the URI of the geospatial data resource that can be represented as tiles and the final part follows the pattern `/tiles`

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

11.5.2. Response

A successful GET response is described in the tileset class

11.6. A tile from a geospatial data resource

A tile resource is a geospatial data resource of a fragment of a more bigger single geospatial data resource that is spatially constrained at the boundaries of the selected tile in a tile matrix set. Details of the operation are described in the tileset class.

Chapter 12. Requirement Class "Map TileSets"

12.1. Overview

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

It will define how to specify the a link to tilesets as a subset mechanism for a map.

NOTE	The concept of map tiles substitutes the WMTS 1.0 concept of tiles. Tiles can be data tiles (e.g. vector tiles) or map tiles (tiles that are rendered using an style)

TBD

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

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

The core of the OGC API - Tiles core draft specification does not mandate the inclusion of an explicit definition of any TileMatrixSet. This draft specification assumes that clients and services know about the eight TileMatrixSets defined in OGC 17-083r2 annex D (or compatible future update of it) and there is no need to define new TileMatrixSets. An extension to the core provides the capability to include definitions of flexible TileMatrixSets that are explicitly defined.

This draft specification assumes that data is organized into one or more geospatial data resources (e.g. the "collections" in OGC API - Features - Part 1: Core [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0]). Geospatial data resources are referenced using URIs.

This document does not specify any requirements for the type of geospatial data resource that should be supported. Provided that the geospatial data resources can be organized into tiles, they can be supported regardless of whether they are features, coverages, a resource that does not represent data per-se (e.g. an annotation) and so forth. The resource path replaces the concept of layer in WMS and WMTS. In this core tiles can be generated from only one geospatial

data resource (tiles that are generated as a combination of geospatial data resources will be defined as an extension).

Accessing the geospatial data resource content (other than as tiles) or its descriptions is out of the scope of this draft specification. If a description of the geospatial data resource exists and it has a mechanism to add links to it, this specification will indicate the need to add a link to the tile description.

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

End of TBD

12.2. 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.

12.2.1. Response

The conformance page mainly consists of a list of links.

Requirement 42	/req/map-tileset/conformance-success
A	If the API has a mechanism to advertise conformance classes, the API SHALL advertise the map tilesets conformance class with a link to http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/map-tileset.

If the server declares also conformity to OGC API - Common or to OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0], then it has to consider the OGC API - Common requirements for declaring conformance, i.e. the use of a the conformance page. In the JSON format the conformance page is an array of links following the link schema defined in the OGC API - Common or in OGC API - Features - Part 1: Core, version 1.0 [http://www.opengis.net/doc/IS/ogcapi-features-1/1.0]. Below is an example fragment of a conformance information page of an API conformant to OGC API - Common and OGC API - Tiles.

Example 10. Conformance Information Page fragment

```
{
  "conformsTo": [
    "http://www.opengis.net/spec/ogcapi-common-1/1.0/conf/core",
    "http://www.opengis.net/spec/ogcapi-common-1/1.0/conf/collections",
    "http://www.opengis.net/spec/ogcapi-maps-1/1.0/conf/core",
    "http://www.opengis.net/spec/ogcapi-maps-1/1.0/conf/dataset",
    "http://www.opengis.net/spec/ogcapi-maps-1/1.0/conf/geodata"
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/tileset"
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/map-tilesets"
  ]
}
```

12.3. Map

This standard does not specify how maps are exposed in the API. Refer to the OGC API - maps Part 1 core, classes dataset or geodata for examples on how to generate maps.

Requirement 43	/req/map-tilesets/desc-links
A	If the API has a mechanism for their maps to expose links to geospatial aspects (e.g. feature items, metadata…), the API SHALL include a `link` with the `href` pointing to every tileset supported by the geospatial data resource that presents a tile aspect of this geospatial data resource and with `rel: "tiles"`.
A	The tileset SHALL be available as a URI that will be composed by three parts: The first part is the URL of the styled map resource (e.g. '…/map/{styleId}' followed by the word '/tiles/', and finally followed by the id of the tile matrix set supported.

Example 11. Fragment of a map description with a links array with one item of the array pointing to a tilesets.

```
{
    "links": [
    ...
    {
        "href":
"http://data.example.com/collections/buildings/map/night/tiles/WebMerca
torQuad",
        "rel": "tiles",
        "type": "application/json",
    }
  ]
}
```

Recommendation 22	/rec/map-tilesets/desc-links
A	If the API has a mechanism for their map resources to expose links to geospatial aspects (e.g. feature items, metadata…), the API SHALL include a `link` with the `href` pointing to a tilesets that presents a tile aspect of this geospatial data resource and with `rel: "tilesets"`.
B	The URI will be composed be the path of the map resources followed by '/tiles'

12.4. Map Tileset

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

12.4.1. Tilesets path

Requirement 44	/req/map-tilesets/operation
A	Every map available as tiles SHALL support an path URL and a HTTP GET operation to retrieve the description of the tiles the API implementation can provide

B	The URI shall be composed by two parts: the initial part is the URI of the styled map (e.g. ⋯/map/{styleID}) that can be represented as tiles and the final part follows the pattern /tiles

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

12.4.2. Response

A successful GET response is described in the tileset class

12.5. A tile from a geospatial data resource

A tile resource is a geospatial data resource of a fragment of a more bigger map that is spatially constrained at the boundaries of the selected tile in a tile matrix set. Details of the operation are described in the tileset class.

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 Core

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

A.1.1. Declaration of conformance classes

A.1.1.1. Response

Abstract Test 1	/ats/core/conformance-success
Test Purpose	Validate that the Conformance Declaration response complies with the required structure and contents.
Requirement	/req/tiles/core/conformance-success
Test Method	1. Validate the response document against OpenAPI 3.0 schema confClasses.yaml 2. Validate that the document includes the conformance class "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/core" 3. Validate that the document lists all OGC API conformance classes that the API implements.

A.1.2. Tiles description

A.1.2.1. Tiles description path

Abstract Test 2	/ats/core/sct-op
Test Purpose	Validate that information about the Tiles can be retrieved from the expected location.
Requirement	/req/tiles/core/sct-op
Test Method	1. Issue an HTTP GET request to the URL {geospatial resource}/tiles 2. Validate that a document was returned with a status code 200

A.1.2.2. Tiles description Link

Abstract Test 3	/ats/core/tc-tile-desc-links
Test Purpose	Validate that the description of the tiles presents a tile aspect of a geospatial resource and with rel: "tiles"
Requirement	/req/tiles/core/tc-tile-desc-links
Test Method	Verify that the response document includes: 1. a link to this response document (relation: self), 2. a link to the response document in every other media type supported by the server (relation: alternate). 3. a link with the href pointing to the description of the tiles that presents a tile aspect of this geospatial resource and with relation: tiles.

A.1.2.3. Response

Abstract Test 4	/ats/core/sct-tmxslink
Test Purpose	Verify that the response to a successful execution to a tiles description contains tileMatrixSetLinks
Requirement	/req/tiles/core/sct-tmxslink

Test Method	1. Validate that the response document contains a property tileMatrixSetLinks
	2. Validate the document against the schema using an JSON Schema validator.

```
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'
```

Abstract Test 5	**/ats/core/sct-tile-examples**
Test Purpose	Verify that the response to a successful execution to a tiles description includes the required tile URI templates.
Requirement	/req/tiles/core/sct-tile-examples

Test Method	1. Verify that the content of the response to a successful execution of a tile description includes at least a link to a tile URI template (rel: item).
	2. Verify that the links provide a URL template with the fragment /tiles followed by the variables {tileMatrixSetId}, {tileMatrix}, {tileRow} and {tileCol}.
	3. Verify that there is a link to a tile URI template for each file format that the server supports (the format is indicated in the 'type' attribute of the link).
	4. Verify that a property 'templated' is part of the link properties to indicate that the link needs to be processed to substitute the templated variables with valid values before being used as a URL to a tile.

A.1.3. A tile from a geospatial resource

A.1.3.1. Operation

Abstract Test 6	**/ats/core/tc-op**
Test Purpose	Validate that a tile can be retrieved from the expected location.
Requirement	/req/tiles/core/tc-op
Test Method	1. Issue an HTTP GET request to the URL with pattern /tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}.
	2. Validate that a content was returned with a status code 200
	3. Validate the contents of the returned feature using test /ats/core/tc-success.

A.1.3.2. Parameter tileMatrixSetId

Abstract Test 7	**/ats/core/tc-tilematrixsetid-definition**
Test Purpose	Validate that the tileMatrixSetId parameters are constructed correctly.

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

Test Method	Verify that the tileMatrixSetId parameter complies with the following definition (using an 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
	``

A.1.3.3. Parameter tileMatrix

Abstract Test 8	/ats/core/tc-tilematrix-definition
Test Purpose	Validate that the tileMatrix parameters are constructed correctly.
Requirement	/req/tiles/core/tc-tilematrix-definition

Test Method	Verify that the tileMatrix parameter complies with the following definition (using an 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'
``` |

A.1.3.4. Parameter tileRow

Abstract Test 9	**/ats/core/tc-tilerow-definition**
Test Purpose	Validate that the tileRow parameters are constructed correctly.
Requirement	/req/tiles/core/tc-tilerow-definition
| Test Method | Verify that the tileRow parameter complies with the following definition (using an OpenAPI Specification 3.0 fragment):<br><br>```
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'
``` |

A.1.3.5. Parameter tileCol

Abstract Test 10	/ats/core/tc-tilecol-definition
Test Purpose	Validate that the tileCol parameters are constructed correctly.
Requirement	/req/tiles/core/tc-tilecol-definition

Test Method	Verify that the tileCol parameter complies with the following definition (using an 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
	``

A.1.3.6. Response

Abstract Test 11	/ats/core/tc-success
Test Purpose	Validate that the response complies with the required format, structure and contents.
Requirement	/req/tiles/core/tc-success
Test Method	1. Validate that a successful execution of the operation is reported with a HTTP status code 200. 2. Validate that the content of that response is consistent with the format requested and represents elements inside or intersecting with the spatial extent of the geographical area of the tile identified by TileMatrixSet, TileMatrix, TileRow and TileCol.

A.2. Conformance Class Root

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

A.2.1. API landing page

A.2.1.1. Response

Abstract Test 12	/ats/root/root-success
Test Purpose	Validate that the API advertises a URI for retrieving tile definitions defined by the service as links to the descriptions paths with rel: tiles.
Requirement	/req/tiles/root/root-success
Test Method	Verify that the API advertises a URI, as links to the descriptions paths with `rel: tiles`, for retrieving tile definitions defined by the service.

A.2.2. Declaration of conformance classes

A.2.2.1. Response

Abstract Test 13	/ats/root/conformance-success
Test Purpose	If the API has a mechanism to advertise conformance classes, validate that the API advertises the capability of generating tiles from multiple collections adding the conformance class for this capability
Requirement	/req/tiles/root/conformance-success

Test Method	Validate that the mechanism the API uses for advertising conformance classes includes the URI `http://www.opengis.net/spec/ogcapi-tiles-1/1.0/conf/root`.

A.2.3. Root tiles description

A.2.3.1. Operation

Abstract Test 14	**/ats/root/ts-op**
Test Purpose	Validate that the server supports retrieval of descriptions of the root tiles
Requirement	/req/tiles/root/ts-op
Test Method	1. Validate that the first part of the URL is the URI of a resource that can be represented as tiles (e.g. /map or simply /) 2. Validate that the second part of the URL follows the pattern `/tiles`. 3. Issue an HTTP GET request to the URL provided for retrieving the description of the root tiles 4. Validate that a document was returned with a status code 200

A.2.3.2. Response

Abstract Test 15	**/ats/root/ts-tile-examples**
Test Purpose	Validate that the response to a tiles request for a root tiles returns a data structure with a link to get tiles representing the resources.
Requirement	/req/tiles/root/ts-tile-examples

Test Method	1. Verify that the response includes at least one link labeled with the relation type `rel: item`
	2. Verify that links with `rel: item` follow a URL template consisting consisting of the variables {tileMatrixSetId}, {tileMatrix}, {tileRow} and {tileCol}
	3. If multiple links are provided with the same URL template, verify that they have a different `type` attribute value (each indicating a different format)

A.2.4. Tiles

A.2.4.1. Operation

Abstract Test 16	**/ats/root/tcs-op**
Test Purpose	Validate that tiles can be retrieved using the URL templates provided by the server.
Requirement	/req/tiles/root/tcs-op
Test Method	1. For every URL template provided by the server, issue an HTTP GET request to the URL, substituting the variables {tileMatrixSetId}, {tileMatrix}, {tileRow} and {tileCol} appropriately.
	2. Validate that responses are returned with a status code 200.
	Repeat the steps above for each format using the values indicated by the `type` attribute.

A.2.4.2. Parameter tileMatrixSetId

Abstract Test 17	**/ats/root/tcs-tilematrixsetid-definition**
Test Purpose	Verify that the operation supports a parameter `tileMatrixSetId`.
Requirement	/req/tiles/root/tcs-tilematrixsetid-definition

Test Method	1. Issue an HTTP GET request with a `tileMatrixSetId` parameter value constructed following schema the below.
| | ```
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
``` |
| | 2. Validate that responses are returned with a status code 200. |

A.2.4.3. Parameter tileMatrix

Abstract Test 18	/ats/root/tcs-tilematrix-definition
Test Purpose	Verify that the operation supports a parameter `tileMatrix`.
Requirement	/req/tiles/root/tcs-tilematrix-definition

Test Method	1. Issue an HTTP GET request with a `tileMatrix` parameter value constructed following schema the below.

	``
	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'
	``

	2. Validate that responses are returned with a status code 200.

A.2.4.4. Parameter tileRow

Abstract Test 19	/ats/root/tcs-tilerow-definition
Test Purpose	Verify that the operation supports a parameter `tileRow`.
Requirement	/req/tiles/root/tcs-tilerow-definition

Test Method	1. Issue an HTTP GET request with a `tileRow` parameter value constructed following schema the below.

```
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'
```

2. Validate that responses are returned with a status code 200.

A.2.4.5. Parameter tileCol

Abstract Test 20	**/ats/root/tcs-tilecol-definition**
Test Purpose	Verify that the operation supports a parameter `tileCol`.
Requirement	/req/tiles/root/tcs-tilecol-definition

Test Method	1. Issue an HTTP GET request with a `tileCol` parameter value constructed following schema the below.

	``
	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
	``
	
	2. Validate that responses are returned with a status code 200.

A.2.4.6. Parameter Resources

Abstract Test 21	**/ats/root/tcs-root-definition**
Test Purpose	Verify that the operation optionally supports a parameter `resources`.
Requirement	/req/tiles/root/tcs-root-definition

Test Method	1. Validate that the operation returns an HTTP 200 code for a `resources` parameter constructed following schema the below.
	
	``
	name: resources
	    in: query
	    required: false
	    style: form
	    explode: false
	    schema:
	      type: array
	      items:
	        type: string
	``
	
	2. Verify that `resources` parameter only contains a comma-separated list of geospatial resource identifiers (collectionId's) or a comma- separated list of full URLs to geospatial resource identifiers.
	
	3. Verify that the geospatial resource identifiers advertise links with type=tiles in the geospatial resource description
	
	4. Verify that the geospatial resource identifiers support the same TileMatrixSetId parameter value

A.2.4.7. Response

Abstract Test 22	/ats/root/tcs-success
Test Purpose	Verify that the response to a tile request is consistent with the spatial extent and media type of resource requested.
Requirement	/req/tiles/root/tcs-success

Test Method	1. Verify that the format of the response is as requested.
	2. Verify that the spatial extent of the contents of the response are inside or intersecting with the spatial extent of the geographical area of the tile identified by TileMatrixSet, TileMatrix, TileRow and TileCol in the request.
	3. Verify that the content of the response complies with the scale denominator represented by the TileMatrix identified.
	4. Validate that the response is returned with a status code 200.

Annex B: Multi-layer Tile Support (Informative)

This draft specification does not impose any limits on the number of data layers that are included in a single tile. The server is therefore allowed to return a tile consisting of multiple data layers, where each individual data layer, or the set of data layers as a whole, may correspond to a collection. Such tiles are referred to as "multi-layer tiles".

Metadata about single or multi-layer tiles may be serialized as JSON, for example using the Mapbox TileJSON [https://github.com/mapbox/tilejson-spec] format version 3. TileJSON conveys information such as the layers found within a tileset, the fields for attribute information, the vector geometry type, the zoom levels as well as a simple URL template for retrieving the tiles themselves. An example TileJSON document is shown in the following section.

B.1. Example TileJSON document

The following TileJSON could be retrieved from a URL such as: https://someserver.ogc.org/tiles/collections/vtp_daraa/tiles/{tileMatrixSetId}?f=application%2Fjson

NOTE	The OGC Vector Tiles Pilot Phase 2 (VTP2) initiative successfully proved that the TileJSON documents could be served from a URL such as: https://someserver.ogc.org/tiles/collections/vtp_daraa/tiles/{tileMatrixSetId}/metadata?f=application%2Fjson

```
{
  "name": "vtp_daraa",
  "scheme": "xyz",
  "tiles": [

"https://someserver.ogc.org/tiles/collections/vtp_daraa/tiles/WebMercatorQuad
/{z}/{y}/{x}?f=application%2Fvnd.mapbox-vector-tile"
  ],
  "center": [
    36.56250000000041,
    34.27502568554792,
    6
  ],
  "bounds": [
    35.8995094299316,
    32.4131851196289,
    36.5781326293945,
    33.1460647583008
  ],
  "vector_layers": [
```

```
    {
      "id": "AgricultureSrf",
      "fields": {
        "OTH": "string",
        "PVH": "number",
        "TSCL": "number",
        "ZI005_FNA": "string",
        "CDR": "string",
        "..." : "..."
      },
      "geometry_type": "polygon"
    },
    {
      "id": "VegetationSrf",
      "fields": {
        "LZN": "number",
        "OTH": "string",
        "PVH": "number",
        "TRE": "integer",
        "..." : "..."
      },
      "geometry_type": "polygon"
    },
    {
      "id": "MilitarySrf",
      "fields": {
        "OTH": "string",
        "WD3": "number",
        "FRT": "integer",
        "FRT3": "integer",
        "FRT2": "integer",
        "ZI005_FNA": "string",
        "..." : "..."
      },
      "geometry_type": "polygon"
    },
    "..."
  ]
}
```

The **draft** schema for TileJSON 3.0.0 is presented below for reference.

```
{
    "name": "TileJSON",
    "type": "object",
```

```json
      "properties": {
          "tilejson": {
              "type": "string",
              "pattern": "\\d+\\.\\d+\\.\\d+\\w?[\\w\\d]*"
          },
          "tiles": {
              "type": "array",
              "items": {
                  "type": "string"
              }
          },
          "vector_layers": {
              "type": "array",
              "items": {
                  "type": "object",
                  "properties": {
                      "id": {
                          "type": "string"
                      },
                      "fields": {
                          "type": "object",
                          "additionalProperties": { "type": "string" }
                      },
                      "description": {
                          "type": "string"
                      },
                      "maxzoom": {
                          "type": "integer"
                      },
                      "minzoom": {
                          "type": "integer"
                      }
                  },
                  "required": [ "id", "fields" ],
                  "additionalProperties": true
              }
          },
          "attribution": {
              "type": "string"
          },
          "bounds": {
              "type": "array",
              "items": {
                  "type": "number"
              }
          },
          "center": {
```

```
            "type": "array",
            "items": {
                "type": "number"
            }
        },
        "data": {
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "description": {
            "type": "string"
        },
        "fillzoom": {
            "minimum": 0,
            "maximum": 30,
            "type": "integer"
        },
        "grids": {
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "legend": {
            "type": "string"
        },
        "maxzoom": {
            "minimum": 0,
            "maximum": 30,
            "type": "integer"
        },
        "minzoom": {
            "minimum": 0,
            "maximum": 30,
            "type": "integer"
        },
        "name": {
            "type": "string"
        },
        "scheme": {
            "type": "string"
        },
        "template": {
            "type": "string"
        },
```

```
            "version": {
                "type": "string",
                "pattern": "\\d+\\.\\d+\\.\\d+\\w?[\\w\\d]*"
            }
        },
        "required": ["tilejson", "tiles", "vector_layers"],
        "additionalProperties": true
    }
```

Annex C: Revision History

Date	Release	Editor	Primary clauses modified	Description
2019-03-21	Template	C. Heazel	all	initial template
2020-04-15	0.0.1	J. Maso	all	Several
2019-04-21	0.0.2	J. Maso	all	Several
2019-05-21	0.0.3	G. Hobona	Annex A	Fixed Conformance Class URI and added abstract tests

Annex D: Bibliography

This standard is deeply inspired in concepts defined in the following documents that preceeded it. This standard offers and alternative interface to fulfill similar tasks included in these references:

- 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)

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

- OGC: OGC 19-014r1: Core Tiling Conceptual and Logical Models for 2D Euclidean Space (2020)

This standard is deeply inspired in concepts and techniques defined in the following OGC API documents. Implementations of this standard can be done also following one of them:

- OGC: OGC 19-072, OGC API - Common Part 1: core

- OGC: OGC 17-069r3, OGC API - Features Part 1

Many ideas standard used in this standard where tested in Testbed 15 and experiences where documented here: * OGC: OGC 19-069, OGC Testbed-15: Maps and Tiles API Engineering Report

The OGC API standards are considering ideas, methods and practices extracted from: * 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 * Mapbox: Mapbox Vector Tiles Specification, https://docs.mapbox.com/vector-tiles/specification/