OGC API - Maps - Part 2

Subsetting by Bounding Box

Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: 2019-03-06

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

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

Version: 0.0.1

Category: OGC® Implementation Specification

Editor: Joan Maso

OGC API - Maps - Part 2: Subsetting by Bounding Box

Warning

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

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

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

License Agreement

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

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

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

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

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

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

Table of Contents

i. Abstract

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

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

ii. Keywords

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

ogcdoc, OGC document, tiling, WMTS

iii. Preface

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

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

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

iv. Submitting organizations

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

Organization name(s)

v. Submitters

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

Name Affiliation

Chapter 1. Scope

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

1.1. Current scope:

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

Chapter 2. Conformance

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

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

The main requirements class is:

- Core.

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

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

C**apture additional requirements classes here**

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

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

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

Chapter 3. References

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

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

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

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

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

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

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

Chapter 4. Terms and Definitions

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

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

4.1. term name

text of the definition

Chapter 5. Conventions

This sections provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

5.1. Identifiers

The normative provisions in this standard are denoted by the URI

http://www.opengis.net/spec/{standard}/{m.n}

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

Chapter 6. Overview

6.1. Evolution from OGC Web Services

OGC Web Service (OWS) standards have historically implemented a Remote-Procedure-Call-over-HTTP architectural style using Extensible Markup Language (XML) for payloads. This was the state-of-the-art when some of the initial versions of OGC Web Services were originally designed in the late 1990s and early 2000s. This architectural style has now a competing RESTful API style that is proposed as an alternative to RPC pattern. A RESTful API style is resource-oriented instead of service-oriented. This OGC API - Maps and Tiles draft specification specifies an API that follows this Web architecture and in particular the W3C/OGC best practices for sharing Spatial Data on the Web as well as the W3C best practices for sharing Data on the Web.

The OGC API – Common draft specification specifies the common kernel of an API approach to services that follows current resource-oriented architecture practices. The draft OGC API - Common specification is the foundation upon which OGC APIs will be built. This common API is to be extended by resource-specific API standards. This draft specification extends OGC API - Common to support Map and Tile resources.

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

- Clear separation between core requirements and more advanced capabilities. This OGC API – Maps and Tiles draft specification presents the requirements that are relevant for almost everyone who wants to share or use Tiled Map Data on a fine-grained level. Additional capabilities that several communities are using today will be specified as extensions to the Core API.
- Technologies that change more frequently are decoupled and specified in separate modules ("requirements classes" in OGC terminology). This enables, for example, the use/re-use of new encodings for spatial data or API descriptions.
- Modularization is not just about a single "service". OGC APIs will provide building blocks that can be reused in APIs in general. In other words, a server supporting the OGC API - Tiles should not be seen as a standalone service. Rather it should be viewed as a collection of API building blocks which together implement Map and Tile capabilities. A corollary for this is that it should be possible to implement an API that simultaneously conforms to conformance classes from the Feature, Coverage, Map, Tiles, and other future OGC Web API standards.

This approach intends to support two types of client developers:

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

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

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

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

6.2. Tiles and maps

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

With time, the concept of a tile has been generalized to other data models such as feature data (some vendors use the expression *vector tiles*) and even to coverage data. This draft specification presents an approach to tiles that can be applied to almost every resource type that returns data representations. If applied in conjunction with the OGC API - Features standard and on top of a feature collection, the expected result is tiled feature data. If applied in conjunction with the OGC API - Maps draft specification and on top of a collection that is transformed into a map by applying a style, the result should be map tiles (usually in PNG or JPEG format).

In this draft specification the OGC API - Tiles is almost fully described. It includes the a core and extensions for defining tile matrix sets, tiles from more that one collection, multi-tiles and multitiles from more than one collection. And info extension is foreseen but not fully developed. In contrast, OGC API - Maps is only partially described based on Testbed-15 requirements. The Maps API is described only to the extent to allow for map tiles to be created on top of a map created by selecting a collection with style or multiple collections with styles. This draft specification contains a section for retrieving a map of an arbitrary number of rows and columns but is not fully formulated. Other extensions for maps are also foreseen. In the future, the WMS SWG could take this document and complete the missing capabilities.

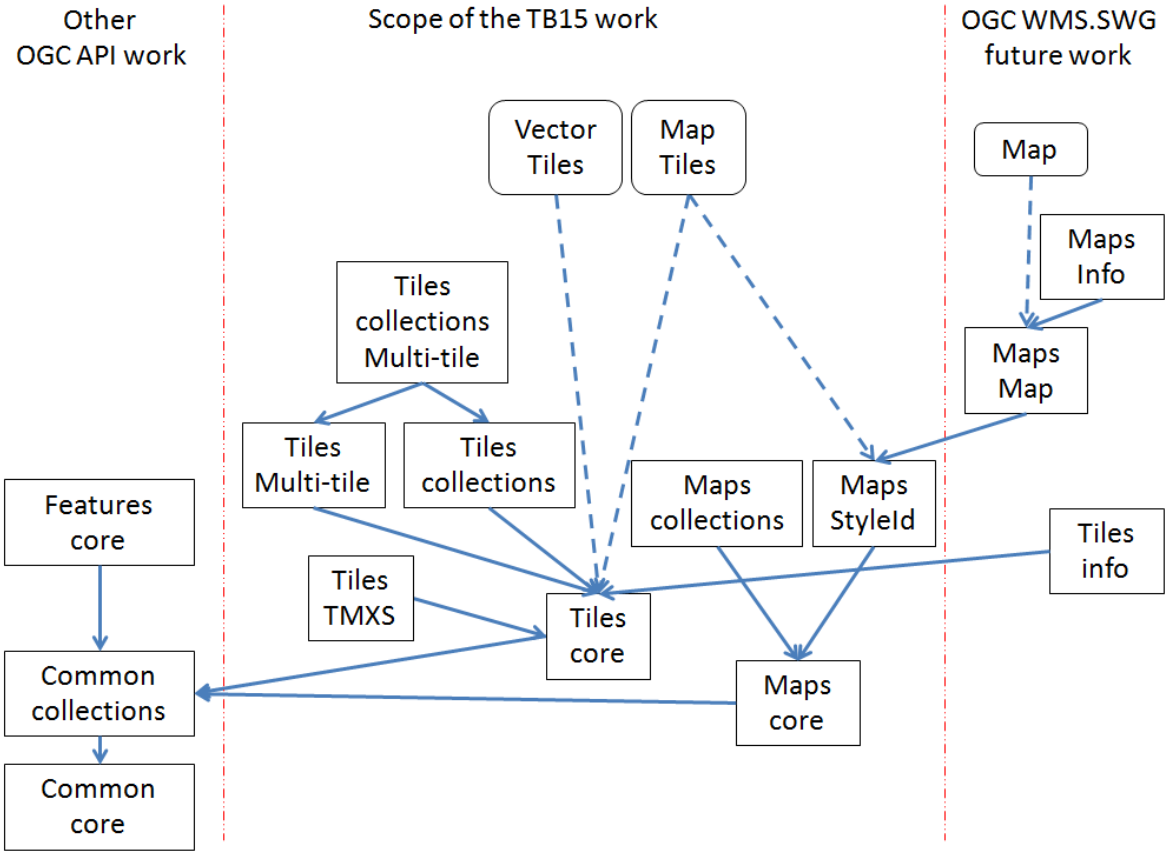

Figure 1. Modular approach in the Maps and Tiles draft specification

6.3. How to approach an OGC API

There are two ways to approach an OGC API.

- Read the landing page, look for links, follow them and discover new links until the desired resource is found
- Read and API definition document that will specify a list of paths to resources.

For the first approach, many resources in the API include links with rel properties to know the reason for this relation. The following figure illustrates does links.

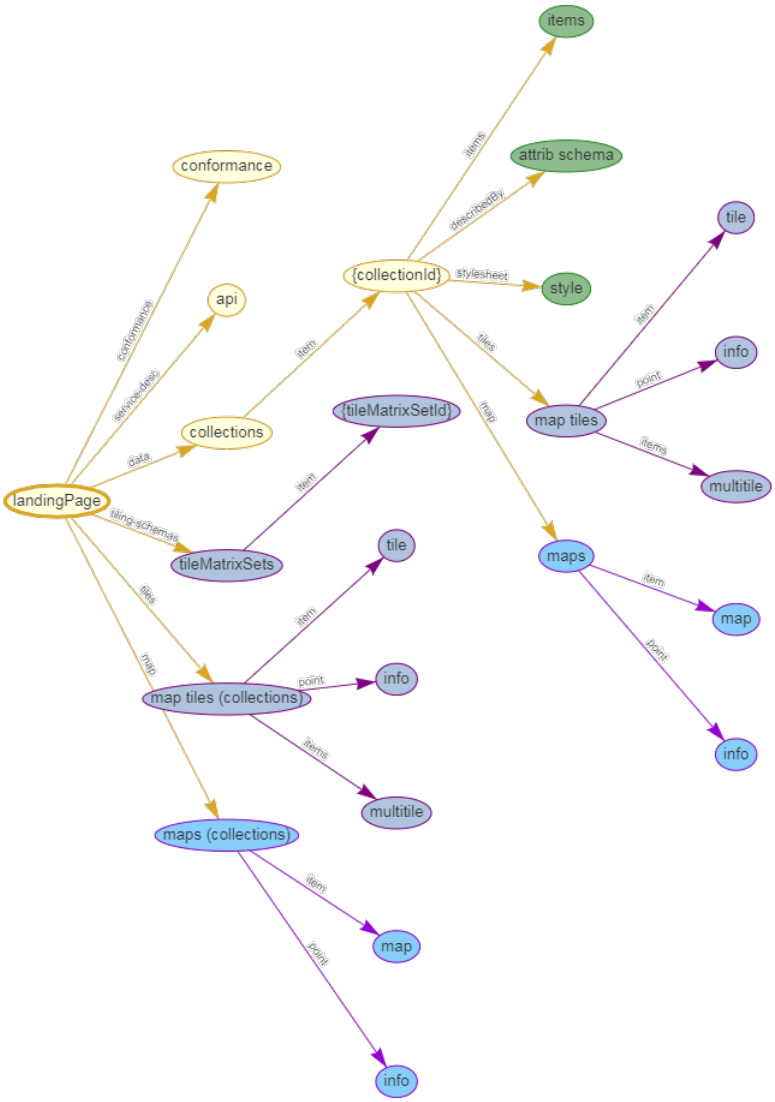

Figure 2. Resources and relations to them via links

For the second approach, the section [OpenAPIExamples] will provide some examples of OpenAPI definition documents that enumerate the paths to get to the necessary resources directly.

Resource name	Common path
Landing page	`/`
Conformance declaration	`/conformance`
Collections	`/collections`
Collection	`/collections/{collectionId}`
Tiling Schemas	`/tileMatrixSets`
Tiling Schema	`/tileMatrixSets/{tileMatrixSetId}`
Tiles	
Vector Tiles description	`/collections/{collectionId}/tiles`

Resource name	Common path
Vector Tiles description from collections	`/tiles`
Vector Tile	`/collections/{collectionId}/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}`
Vector tile collections[1]	`/map/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}`
Vector Multi-tiles	`/collections/{collectionId}/tiles/{tileMatrixSetId}`
Vector Multi-tiles collections[1]	`/tiles/{tileMatrixSetId}`
Map tiles	
Map tiles description	`/collections/{collectionId}/map/`
Map tiles description collections[1]	`/map/tiles`
Map tile	`/collections/{collectionId}/map/{styleId}/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}`
Map tile collections[1]	`/map/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}`
Map tile multi-tiles	`/collections/{collectionId}/map/{styleId}/tiles/{tileMatrixSetId}`
Map tile multi-tiles collections[1]	`/map/tiles/{tileMatrixSetId}`
Maps	
Maps description	`/collections/{collectionId}/map`
Maps description collections[1]	`/map`

Table 1. Overview of resources and common direct links defined in the API

[1]: In first column of the table, the word "collections" means "from more than one collection"

Chapter 7. Requirement Class "Map BBox"

7.1. Overview

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

This extension describes how to subset a map by specifying a set of parameters that will determine its resolution (width, height, boundingbox and CRS).

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. OGC API - Common already requires some links.

Requirement 7	/req/core/conformance-success
A	If the API has a mechanism to advertise conformance classes, the API SHALL advertise the maps bbox conformance class with a link to http://www.opengis.net/spec/ogcapi-maps-1/1.0/conf/bbox.

In the conformance page (typically in JSON format) the links follow the link schema defined in the OGC API - Common. The following is an example fragment of the response of an implementation of the OGC API – Maps draft specification with the maps extension conformance information page.

Example 1. Conformance Information Page fragment

```
{
  "conformsTo": [
  [
    "http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core",
    "http://www.opengis.net/spec/ogcapi-common-1/1.0/req/collections",
    "http://www.opengis.net/spec/ogcapi-maps-1/1.0/req/core"
    "http://www.opengis.net/spec/ogcapi-maps-1/1.0/req/bbox"
  ]
}
```

7.3. Map subset by bounding box

7.3.1. Operation

The core of this standard defines how to add a {styleId} parameter to a map to make retrievable as a subset. This standard specifies the parameters needed to subset a map an retrieve fragments of it.

An example of a resource that can be retrieved as map subsets is a feature collection (/collections/{collectionId}/map/{styleId}).

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. In addition the parameters specified by the core, other parameters should be added.

7.3.2. Parameter crs

This parameter indicates the CRS of the data rendered in the map as well as the CRS used by the parameter bbox.

Requirement 8	**/req/bbox/crs-definition**

A	The map operation SHALL support a parameter `crs` with the characteristics defined in the OpenAPI Specification 3.0 fragment
	
	`crsId:`
	`  name: crs`
	`  in: query`
	`  description: A URI of the coordinate reference system of the bbox and the map subset response. A list of all available CRS values can be found under the map description resource.`
	`  required: true`
	`  schema:`
	`  type: string`
	`  example: http://www.opengis.net/def/crs/OGC/1.3/CRS84`

7.3.3. Parameter bbox

This parameter indicates the size of the bbox of the subset of the data (in the crsId coordinates). In this OPC API extension the bbox parameter values are defined considering the exterior edges of the pixels of the viewport defined by the width and height. In other words, the relation of the Bounding Box to the map pixel matrix is that the bounding box goes around the "outside" of the pixels of the map rather than through the centers of the map's border pixels (as many times done in coverages). In this context, individual pixels represent a cell area on the ground.

Requirement 9	**/req/bbox/width-definition**

A	The map operation SHALL support a parameter `bbox` with the characteristics defined in the OpenAPI Specification 3.0 fragment
	
	`bbox:`
	`  name: bbox`
	`  in: query`
	`  description:`
	`  Bounding box of the rendered map.The bounding box is provided as four coordinates`
	
	`  * Lower left corner, coordinate axis 1`
	`  * Lower left corner, coordinate axis 2`
	`  * Upper right corner, coordinate axis 1`
	`  * Upper right corner, coordinate axis 2`
	
	`  The coordinate reference system and axis order of the values are indicated in the `crs` parameter.`
	`  required: false`
	`  schema:`
	`  type: array`
	`  minItems: 4`
	`  maxItems: 6`
	`  items:`
	`  type: number`
	`  format: double`
	`  style: form`
	`  explode: false`

B	`bbox` SHALL be a comma separated list of floating point numbers. The first two numbers are the coordinates of the lower left corner. The last two are the coordinates of the upper right corner. The axis order is determined by the `crs` parameter value. For example in CRS84 the order is long_min, lat_min, long_max, lat_max.
C	The `bbox` parameter values SHALL not exceed the optional resource limits provided in the map description document under the `crsSpatialExtent` element that corresponds to the `crs` parameter value.

NOTE	This parameter use the comma (",") as the separator between the coordinates. Additional white space will not be used to delimit list items. The comma character should not be escaped (IETF RFC 2396).

7.3.4. Parameter width

The subset of the map will specify the size of the viewport in rows and columns. If the format of the response is a raster image, the size of the image will match the requested width and height.

Requirement 10	**/req/bbox/width-definition**

A	The map operation SHALL support a parameter `width` with the characteristics defined in the OpenAPI Specification 3.0 fragment
	
	`width:`
	`  name: width`
	`  in: query`
	`  description:`
	`  Width of the viewport to present the response (the map subset).`
	`  required: false`
	`  style: form`
	`  explode: false`
	`  schema:`
	`  type: number`
	`  default: 256`

B	`width` SHALL be a positive integer number indicating the horizontal size (columns) of the viewport where the response will be presented.
C	The value of the `width` SHALL no excide the capabilities of the server provided in the map description document under the `maxWidth` property.

7.3.5. Parameter height

Requirement 11	**/req/bbox/heigth-definition**

A	The map operation SHALL support a parameter `height` with the characteristics defined in the OpenAPI Specification 3.0 fragment
	
	`height:`
	`  name: height`
	`  in: query`
	`  description:`
	`  Height in pixels of the viewport to present the response (the map subset).`
	`  required: false`
	`  style: form`
	`  explode: false`
	`  schema:`
	`  type: number`
	`  default: 256`

B	`height` SHALL be a positive integer number indicating the vertical size (rows) of the viewport where the response will be presented.
C	The value of the `height` SHALL no excide the capabilities of the server provided in the map description document under the `maxHeight` property.

7.3.6. Response

A successful response to a map bbox GET operation will be consistent with the media type of resource requested. This draft specification does not impose any media type or file format and maps response may be in JPEG, PNG or other appropriate format (including vector based formats such as KML or SVG).

Requirement 12	**/req/tileset/tc-success**
A	A successful execution of the maps subset by bbox 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 the crsId requested and represent elements inside or intersecting with the spatial extent of the geographical area of the map identified by bbox.
C	In the case of raster image media type (e.g. image/jpeg or image/png) the number of columns and rows SHALL be exactly the ones specified in the width and height parameters respectively.

Permission 2	**/per/bbox/map-encoding**
A	This draft specification does not impose any media type on the encoding of a map response. For raster maps it MAY be a JPEG, PNG or other format. For vector formats it could by KML or SVG.

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

Recommendation 2	**/rec/bbox/map-success-scale**
A	The resolution of the content represented in the map SHOULD be consistent with the width and height of the viewport indicated. In case of vector data (e.g. KML or SVG) the geometries should be simplified to match the resolution of the viewport (raster responses are simplified to match the width and height).

7.3.7. 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 `crsId` is not available by the server for this resource or the parameters values `bbox`, `width`, `height` are out-of-range, or the map is not provided due to lack of data in the area, the status code of the response will be `404`.

Annex A: Conformance Class Abstract Test Suite (Normative)

NOTE Ensure that there is a conformance class for each requirements class and a test for each requirement (identified by requirement name and number)

A.1. Conformance Class A

A.1.1. Requirement 1

Test id:	/conf/conf-class-a/req-name-1
Requirement:	/req/req-class-a/req-name-1
Test purpose:	Verify that…
Test method:	Inspect…

A.1.2. Requirement 2

Annex B: Revision History

Date	Release	Editor	Primary clauses modified	Description
2019-03-21	Template	C. Heazel	all	initial template