

# OGC Testbed-15:Images and Changeset API Engineering Report

# Table of Contents

1. Scope .....	8
2. Conformance .....	9
3. References .....	10
4. Terms and definitions .....	11
4.1. Abbreviated terms .....	12
5. Conventions .....	13
5.1. Identifiers .....	13
5.2. Link relations .....	13
5.3. Use of HTTPS .....	14
5.4. HTTP URIs .....	14
5.5. API definition .....	14
5.5.1. General remarks .....	14
5.5.2. Role of OpenAPI .....	14
5.5.3. References to OpenAPI components in normative statements .....	15
5.5.4. Paths in OpenAPI definitions .....	15
5.5.5. Reusable OpenAPI components .....	15
6. Overview .....	16
6.1. Evolution from OGC Web Services .....	16
6.2. Images, Tiles and Changesets .....	16
6.3. How to approach the OGC API Images .....	16
7. Requirements Class "Images Core" .....	19
7.1. Overview .....	19
7.2. General .....	20
7.3. API landing page .....	20
7.3.1. Response .....	20
7.4. Declaration of conformance classes .....	21
7.4.1. Response .....	21
7.5. Collections .....	21
7.6. Collection .....	22
7.6.1. Collection Links .....	22
7.7. Image set description .....	22
7.7.1. Operation .....	22
7.7.2. Parameter collectionId .....	23
7.7.3. Parameters limit and offset .....	23
7.7.4. Parameters bbox and datetime .....	23
7.7.5. Response .....	23
7.8. Image description .....	25
7.8.1. Operation .....	25

7.8.2. Parameter collectionId .....	26
7.8.3. Parameter imageId .....	26
7.8.4. Response .....	26
8. Requirements Class "Images Transactional" .....	32
8.1. Overview .....	32
8.2. API landing page .....	32
8.2.1. Response .....	32
8.3. Declaration of conformance classes .....	32
8.3.1. Response .....	32
8.4. Image upload .....	33
8.4.1. Operation .....	33
8.4.2. Parameter collectionId .....	34
8.4.3. Payload .....	34
8.4.4. Response .....	34
8.4.5. Operation .....	35
8.4.6. Parameter collectionId .....	35
8.4.7. Parameter imageId .....	35
8.4.8. Request payload .....	35
8.4.9. Response .....	36
8.4.10. Operation .....	36
8.4.11. Parameter collectionId .....	36
8.4.12. Parameter imageId .....	36
8.4.13. Response .....	36
9. Requirement Class "Changeset core" .....	37
9.1. Overview .....	37
9.2. General .....	38
9.3. API landing page .....	38
9.3.1. Response .....	38
9.4. Declaration of conformance classes .....	38
9.4.1. Response .....	38
9.5. Resource retrieval .....	39
9.5.1. Headers .....	39
9.6. Changeset extra parameters .....	39
9.6.1. Parameter Checkpoint .....	40
9.6.2. Parameter priority .....	40
9.6.3. Parameter changeSetType .....	41
9.7. Changeset response .....	41
9.7.1. Changeset response with changes .....	42
9.7.2. Changeset response with no changes .....	45
10. Requirement Class "Changeset tiles" .....	46
10.1. Overview .....	46

10.2. API landing page .....	46
10.3. Declaration of conformance classes .....	46
10.3.1. Response .....	46
10.4. Multi-tiles operation returning a changeset .....	47
10.4.1. Parameter priority .....	47
10.4.2. Parameter multiTileType .....	48
10.4.3. Parameter changeSetType .....	48
10.5. Changeset response .....	48
Appendix A: Abstract Test Suite .....	50
Appendix B: STAC Schemas and examples .....	51
B.1. STAC collection schema .....	51
B.2. STAC item schema extended for images .....	55
Appendix C: OpenAPI Domain .....	62
C.1. OpenAPI Domain images .....	62
C.2. OpenAPI Domain changset .....	72
Appendix D: OpenAPI Examples .....	78
D.1. OpenAPI Images example .....	78
D.2. OpenAPI Images example .....	87
Appendix E: Revision History .....	100
Appendix F: Bibliography .....	101

Publication Date: 2020-01-08

Approval Date: 2019-11-22

Submission Date: 2019-10-30

Reference number of this document: OGC 19-070

Reference URL for this document: <http://www.opengis.net/doc/PER/t15-D016>

Category: OGC Public Engineering Report

Editor: Joan Maso Pau

Title: OGC Testbed-15:Images and ChangesSet API Engineering Report

---

## **OGC Public Engineering Report**

### **COPYRIGHT**

Copyright © 2020 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

### **WARNING**

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Public Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

## 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.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

## **i. Abstract**

The OGC API – Images and Changeset draft specification addresses the use case of an OGC API tile server that serves image tiles and a client that portrays the result as a set of images. The tile server uses a set of images (e.g. a set of remote sensing satellite scenes or a set of drone pictures) in the backend and they are also accessible by an API - Images. The source images can be updated and therefore the tile server also needs to be able to deliver only the tiles that have changed. The draft specification is divided into two independent parts that can be used in broader scenarios:

- The OGC API – Images: Enables managing (retrieving, creating and updating) sets of images that are georeferenced. The images does not follow any tile scheme, and can partially or totally overlap. The API enables a mosaicking use case (where the imagery is combined in a single bigger “picture”) but could also serve a use case in which a moving camera is taking pictures at locations along a route and then stores the images as a single collection.
- The Changeset filter: Enables filtering a request to a data service in a way that only recent changes are delivered. It can be applied to OGC API that provide access to data and in particular to the OGC API tiles.

The OGC API – Images simplifies the creation and maintenance of sets of images that can then be exposed and retrieved by other OGC API’s, such as OGC API – Coverages.. The use of the Changeset filter helps keep clients synchronized with changes to the source content on servers while also minimizing the bandwidth necessary for the synchronization.

This draft specification covers the following conformance classes:

- The image core describes how to retrieve images from an image set.
- The image transaction describes how to create or update an image in an existing image set.
- The changeset core describes how to filter data requests using a checkpoint and how the changeset is encoded
- The changeset tiles details the changeset encoding in the case of a multi-tiles request.

The OWS Common SWG is expected to discuss and achieve consensus on the changeset requirements classes and integrate them into a building bloc that every data OGC API could use. The Web Coverage Service (WCS) SWG is expected to consider the OGC API – Images requirements classes to provide an easy solution for creating and updating sets of images that are the backend content for a coverage server.

### **i.i Executive Summary**

This draft specification is addressing the case of a tile server that serves tiles that portrays the result of mosaicking a set of images. This images can be updated from time to time and the tile server needs to be able to deliver only the tiles that has changed. The draft specifcaiton can be divided into two independent parts that can be used in broader scenarios:

- The OGC API images: it allows managing (retrieving, creating and updating) sets of images that are georeferenced. It serves the mosaicking use case but it could also serve a case where a moving camera is taking pictures at some points in a route and stores the images as a single collection.



- The ChangeSet filter: it allows filtering a request to a data service in a way that only recent changes are delivered.

The OGC API images simplifies the creation and maintenance of sets of images that can then be exposed and retrieved by other OGC API's such as OGC API coverages, OGC API coverates etc. The use of ChangeSet filter helps on keeping clients synchronized with changes in servers, minimizing the bandwidth necessary for the synchronization.

This draft specification covers the following conformance classes:

- The *image core* describes how to retrieve images from an image set.
- The *image transaction* describes how create or update an image in an existing image set.
- The *tiles info* proposes a mechanism to provide extra information about one point in a tile. It requires further elaboration.
- The *changeset core* describes how to filter data requests using a checkpoint and how the changeset is encoded
- The *changeset tiles* describes the particularities of the changeset encoding in the case of request to multi-tiles.

The OWS Common.SWG is expected to take case of the changeset requirements classes and integrate them into a building bloc that every data OGC API could use.

The WCS.SWG is expected to consider the OGC API - images requirements classes to provide an easy solution for creating an updating sets of images that aliment a coverage server.

## ii. Keywords

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

ogcdoc, OGC document, tiling, WMTS, image, API, updates, synchronization

## iii. Preface

The content of this OGC Engineering Report (ER) focuses on the use case in which a service provides access to a set of images that are part of a spatiotemporal series, such as a remote sensing product distributed as granules. These images are regularly updated with new imagery such as from an active remote sensing platform that produces updated images of the Earth. To fulfill requirements for the above use case, an image service uses the draft OGC API - Tiles specification, which is an evolution of the OGC Web Map Tile Service (WMTS), to serve this image set. The tile client maintains a cache of the previously accessed images and therefore only needs to retrieve the tiles that have been updated since the last interaction with the server.

To solve the above use case, two services are proposed: A tile service with an image service that is internally synchronized. An image set is identified by a collection id. An image set is a list of images and each one having an image id and it is exposed by an image server through an OGC API - Images. A Tile server also exposes the image set as tiles associated to a layer name. Both views of the same dataset are related when the image set collection id and the tile server layer name are the same. Checkpoint ids are used to notify the status of the client cache to the server. The Checkpoint id acts as a filter in retrieval queries, making the server respond with only the tiles updated since the last

interaction. In other words, it allows the server to recognize the client status and to send it the changes to update the status to the current server status. The way the server “recognizes the status of the client” is up to a server developer: it might involve the need to maintain server replicas of content of previous communications to the client. Another way to implement the same functionality is that the service documents (audits) every change in the resource(s) it maintains and assign a checkpoint to each change independently of the client interaction.

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](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):

UAB-CREAF

#### **v. Submitters**

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

Joan Masó, CREAM

#### **vi. Document contributor contact points**

All questions regarding this document should be directed to the editor or the contributors:

#### **Contacts**

<b>Name</b>	<b>Organization</b>	<b>Role</b>
Joan Maso Pau	Universitat Autònoma de Barcelona (CREAF)	Editor

The draft specification was based on (and sometimes reproduces) previous work of the following people:

#### **Contacts**

Name	Organization	Role
Keith Pomakis	Cubewerx	The OGC API -Images reproduces the behavior of his OGC API - Images design and implementation
Peter Vretanos	Cubewerx	The ChangeSet building is based on his previous idea of a checkpoint resource, modified here as a checkpoint filter parameter.

**vii. Foreword**

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.

# Chapter 1. Scope

This OGC Testbed-15 Engineering Report (ER) provides an approach to a transactional Web Map Tiling Service 2.0 (WMTS-T) by combining an Image Service to manage (retrieve, add, modify or delete) a set of images in a server with a WMTS service that is able to communicate updates within a specified timeframe. Both the Image Service and the WMTS are internally connected. WMTS is not transactional but the transactional capabilities are provided by the Image Service.

This ER presents a proof of concept developed in Testbed-15 and presented in this document in the form of a draft specification. The draft specification is written in the context of the OGC Application Programming Interface (API) effort and abandons the OGC web services style (KVP, SOAP, etc.) to adopt the new OGC Web API style whereby services use a resource-oriented architecture along with the standard web verbs and an OpenAPI description document. The draft specification is documented for consideration by the OGC Standards Program.

# Chapter 2. Conformance

If this draft specification progresses within the Standards Program, this section would be populated by the Web Map Service (WMS) Standards Working Group (SWG).

# Chapter 3. References

The following normative documents are referenced in this document.

*NOTE: Only normative standards are referenced here, e.g. OGC, ISO or other SDO standards. All other references are listed in the bibliography. Example:*

- [OGC: OGC 06-121r9, OGC® Web Services Common Standard \(2010\)](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact\_id=38867&version=2]
- [OGC: OGC 07-057r7, OGC® OpenGIS Web Map Tile Service Implementation Standard \(2010\)](http://portal.opengeospatial.org/files/?artifact_id=35326) [http://portal.opengeospatial.org/files/?artifact\_id=35326]
- [OGC: OGC 17-069r3, OGC API - Features - Part 1: Core \(2019\)](http://docs.opengeospatial.org/DRAFTS/17-069r3.html) [http://docs.opengeospatial.org/DRAFTS/17-069r3.html]
- [OGC: OGC 17-089r1, OGC® Web Coverage Service \(WCS\) Interface Standard - Core, version 2.1 \(2018\)](http://docs.opengeospatial.org/is/17-089r1/17-089r1.html) [http://docs.opengeospatial.org/is/17-089r1/17-089r1.html]
- [OGC: OGC 17-083r2, OGC Two Dimensional Tile Matrix Set \(2019\)](http://docs.opengeospatial.org/is/17-083r2/17-083r2.html) [http://docs.opengeospatial.org/is/17-083r2/17-083r2.html]
- [IETF: RFC 2616, HTTP/1.1 \(1999\)](https://tools.ietf.org/html/rfc2616) [https://tools.ietf.org/html/rfc2616]
- [IETF: RFC 2818, HTTP over TLS \(2000\)](https://tools.ietf.org/html/rfc2818) [https://tools.ietf.org/html/rfc2818]
- [IETF: RFC 3339, Date and Time on the Internet: Timestamps \(2002\)](https://tools.ietf.org/html/rfc3339) [https://tools.ietf.org/html/rfc3339]
- [IETF: RFC 8288, Web Linking \(2017\)](https://tools.ietf.org/html/rfc8288) [https://tools.ietf.org/html/rfc8288]

## NOTE

This draft specification produced in the Testbed-15 project assumes that an OGC API - Common standard exists. This was a necessary assumption in the Testbed-15 project. However, at the time of finalizing this engineering report an initial draft specification had been produced in the OWS Common SWG. The authors of this document are assuming that some parts of the OGC API - Features standard will form part of a new OGC API - Commons in a near future with almost no variation, but with the text generalized to cover resources types other than features).

## NOTE

This draft specification has normative references to the SpatioTemporal Asset Catalog (STAC) specification that has not been vetted and approved by a standards development organization such as the OGC [1]. Actually, the STAC specification is still rapidly evolving and the latest version can be found in <https://github.com/radiantearth/stac-spec>. As a mitigation, this draft specification reproduces the schemas of the version 0.8. The opinion of the authors of this draft specification is that the OGC API - Images core could be written in a way that is independent from the STAC encoding and an extension can be added to support the STAC encoding. In the Testbed-15 there was no time or the need to do it but a future SWG should consider this as an option.

# Chapter 4. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact\_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

- **changeset**

an atomic collection of changes to a set or resource in a repository. A changeset contains all recorded local modified resources that lead to current status of the repository from a previous status. The previous status is identified by a checkpoint ID.

- **checkpoint**

in a situation where the server can update resources and collections with new information, a moment in time when the server exchanges resource information with the client and the client becomes synchronized with the server (i.e both client and server have a common representation of those resources). A checkpoint is represented by a checkpoint ID that can later be used by a client to request an update of the resources (i.e. a changeset).

- **coordinate reference system**

coordinate system that is related to an object by a datum (source: ISO 19111:2019)

**NOTE** Geodetic and vertical datums are referred to as reference frames.

**NOTE** For geodetic and vertical reference frames, the object will be the Earth. In planetary applications, geodetic and vertical reference frames may be applied to other celestial bodies.

- **dataset**

collection of data (source: ISO 19115)

**NOTE** [DCAT] defines 'dataset' as a collection of data, published or curated by a single agent, and available for access or download in one or more formats. The use of 'collection' in the definition from [DCAT] is broader than the use of the term collection in this specification.

- **distribution**

represents an accessible form of a **dataset** [DCAT]

EXAMPLE: a downloadable file, an RSS feed or an API.

- **collection**

a set of **elements** from a **dataset**

**NOTE** In this specification, 'collection' is used as a synonym for 'feature collection'. This is done to make, for example, URI path expressions shorter and easier to understand for those that are not geo-experts.

- **element**

entities that are part of a collection

- **image set**

list of images with many aspects in common that allows them to be presented as a single one (e.g. as an image mosaic). Commonly they share the same CRS, the same topic (in remote sensing they will share the same sensor and level of processing) and they are from similar times.

- **mosaic**

a continuous image representation that results from combining all images in an image set

- **portrayal**

presentation of information to humans (source: ISO 19117)

- **tile**

a geometric shape with known properties that is the result of the tiling (tessellation) of a plane. A tile consists of a single connected "piece" without "holes" or "lines" (topological disc).

- **tile matrix**

a grid tiling scheme that defines how space is partitioned into a set of conterminous tiles at a fixed scale (source: OGC 17-083r2).

NOTE A tile matrix constitutes a tessellation of the space that resembles a matrix in a 2D space characterized by a matrix width (columns) and a matrix height (rows).

- **tile matrix set**

a tiling scheme composed by collection of tile matrices defined at different scales covering approximately the same area and has a common coordinate reference system (source: OGC 17-083r2).

- **Web API**

API using an architectural style that is founded on the technologies of the Web [\[DWBP\]](#)

**NOTE**

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

## 4.1. Abbreviated terms

- API Application Programming Interface
- CRS Coordinate Reference System
- OWS OGC Web Services
- STAC SpatioTemporal Asset Catalog
- WCS Web Coverage Service
- WMTS Web Map Tile Service



# Chapter 5. Conventions

## 5.1. Identifiers

The normative provisions in this draft specification are denoted by the URI <http://www.opengis.net/spec/ogcapi-images-1/1.0> and <http://www.opengis.net/spec/ogcapi-changeset-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) is used.

The following registered link relation types are used in this document.

- **alternate**: Refers to a substitute for this context.
- **collection**: The target IRI points to a resource which represents the collection resource for the context IRI.
- **describedBy**: Refers to a resource providing information about the link's context.
- **item**: The target IRI points to a resource that is a member of the collection represented by the context IRI.
- **next**: Indicates that the link's context is a part of a series, and that the next in the series is the link target.
- **license**: Refers to a license associated with this context.
- **prev**: Indicates that the link's context is a part of a series, and that the previous in the series is the link target.
  - This relation is only used in examples.
- **self**: Conveys an identifier for the link's context.
- **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.

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

- **items**: Refers to a resource that is comprised of members of the collection represented by the link's context.
- **conformance**: Refers to a resource that identifies the specifications that the link's context conforms to.

- **data:** Indicates that the link's context is a distribution of a dataset that is an API and refers to the root resource of the dataset in the API.

Each resource representation includes an array of links. Implementations are free to add additional links for all resources provided by the API. For example, an **enclosure** link could reference a bulk download of a collection. Or a **related** link on a feature could reference a related feature.

## 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 RFC2818 (HTTPS), not RFC2616 (HTTP).

## 5.4. HTTP URIs

This document does not restrict the lexical space of URIs used in the API beyond the requirements of the RFC2616 (HTTP) and RFC3986 (URI Syntax) IETF RFCs. If URIs include reserved characters that are delimiters in the URI subcomponent, these have to be percent-encoded. See Clause 2 of RFC3986 for details.

## 5.5. API definition

### 5.5.1. General remarks

Good documentation is essential for every API so that developers can more easily learn how to use the API. In the best case, documentation will be available in HTML and in a format that can be processed by software to connect to the API.

This draft specification specifies requirements and recommendations for APIs that share images and to specify changesets and that want to follow a standard way of doing so. In general, APIs will go beyond the requirements and recommendations stated in this draft specification - or other parts of the OGC API family of standards - and will support additional operations, parameters, etc. that are specific to the API or the software tool used to implement the API.

### 5.5.2. Role of OpenAPI

This document uses OpenAPI 3.0 fragments as examples and to formally state requirements. However, using OpenAPI 3.0 is not required for implementing a server.

Therefore, the *Core* requirements class only requires that an API definition is provided and linked from the landing page.

In this document, fragments of OpenAPI definitions are shown in YAML (YAML Ain't Markup Language) since YAML was designed to be easier to read than JavaScript Object Notation (JSON) and is typically used in OpenAPI editors. YAML is described by its authors as a human friendly data serialization standard for all programming languages.

### 5.5.3. References to OpenAPI components in normative statements

Some normative statements (requirements, recommendations and permissions) use a phrase that a component in the API definition of the server must be "based upon" a schema or parameter component in the OGC schema repository.

In the case above, the following changes to the pre-defined OpenAPI component are permitted.

- If the server supports an Extensible Markup Language (XML) encoding, `xml` properties may be added to the relevant OpenAPI schema components.
- The range of values of a parameter or property may be extended (additional values) or constrained (if a subset of all possible values is applicable to the server). An example for a constrained range of values is to explicitly specify the supported values of a string parameter or property using an enum (enumeration).
- The default value of a parameter may be changed or added unless a requirement explicitly prohibits this.
- Additional properties may be added to the schema definition of a Response Object.
- Informative text may be changed or added, like comments or description properties.

For API definitions that do not conform to the OpenAPI Specification 3.0, the normative statement should be interpreted in the context of the API definition language used.

### 5.5.4. Paths in OpenAPI definitions

All paths in an OpenAPI definition are relative to a base URL of the server.

*Example 1. URL of the OpenAPI definition*

If the OpenAPI Server Object looks like this:

```
servers:  
- url: https://dev.example.org/  
  description: Development server  
- url: https://data.example.org/  
  description: Production server
```

The path `"/mypath"` in the OpenAPI definition of a Web API would be the URL <https://data.example.org/mypath> for the production server.

### 5.5.5. Reusable OpenAPI components

Reusable components for OpenAPI definitions for implementations of OGC APIs are referenced from this document.

# Chapter 6. Overview

## 6.1. Evolution from OGC Web Services

This OGC Testbed-15 Engineering Report defines a draft specification that formalizes the approach that the Testbed-15 participants agreed to for providing a Web Map Tiling Service – transactional capability. In Testbed-15, the WMTS was based on a set of images (e.g. remote sensing scenes) that were the source for the creation of the tiles served by a WMTS implementation. The content of the tiles can change over time. This could result from new scenes being provided by some remote sensing platform. Instead of making the WMTS service (and standard) capable of accepting new tiles as updates (WMTS transactional), the Testbed 15 participants instead decided to opt for pairing a tile service with an image service that are internally synchronized. An image set is exposed by the image server as a list of images that has an id and by the tile server as tiles associated to a layer with the same id. The concept of an image service is new to the emerging suite of OGC APIs.

## 6.2. Images, Tiles and Changesets

The next two sections specify the OGC API - Images draft specification. The first section describes the core of the service: The operations related to service discovery, API description, collection description, image sets description and operations for retrieving images. The second section describes the mechanisms for adding, updating and deleting images of an image set.

These sections are followed with a discussion and description of the changeset building block. First, a changeset workflow that can be applied to any data service is specified. The draft specification specifies a filter parameter to retrieve only the elements that have changed in a collection. The second following section specifies the way the changeset filter is applied to the OGC API tiles.

## 6.3. How to approach the OGC API Images

From a client implementation point of view, there are mainly 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 the API definition document (for example an OpenAPI document) that specifies a list of paths to resources.

Since the first approach requires visiting several resources until the desired resource is finally reached, it is expected to be slower than the second approach. The first approach mimics how the web works and it is only recommended to implementers of very generic exploratory clients. In this approach, many resources in the API include links with rel properties to show the reason for the relation. The following figure illustrates the links.

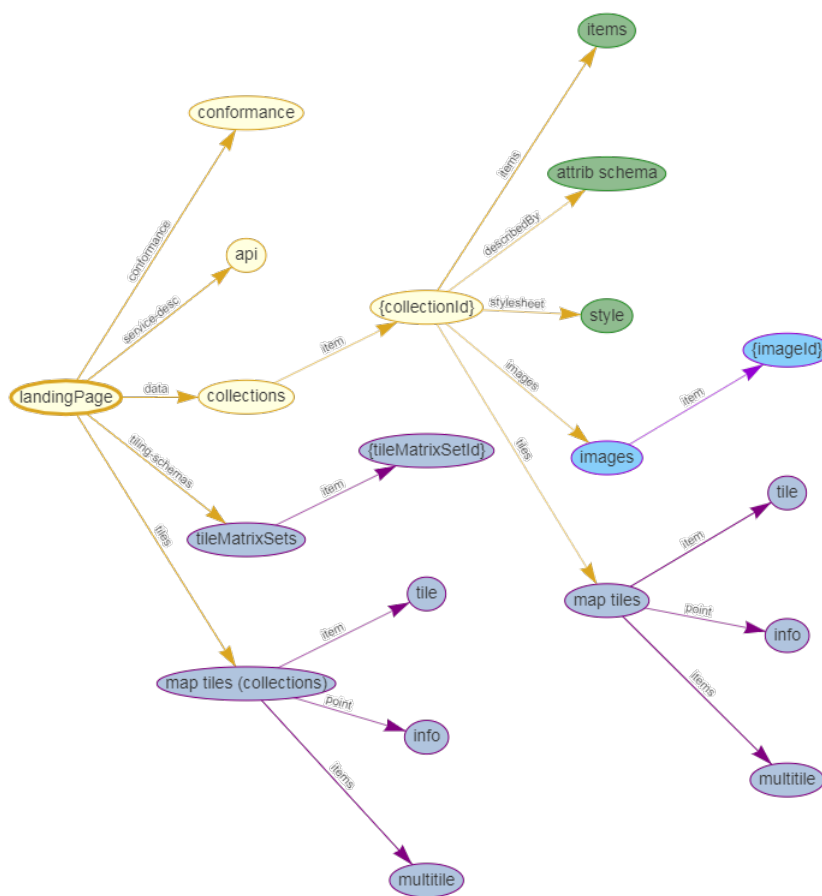


Figure 1. Resources and relations to them via links

For the second approach, the section [OpenAPI Examples](#) will provide some examples of OpenAPI definition documents that enumerate the paths to get to the necessary resources directly [2].

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

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

Resource name	Common path
Map tiles description collections <sup>1</sup>	/map/tiles
Map tile	/collections/{collectionId}/map/{styleId}/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}
Map tile collections <sup>1</sup>	/map/tiles/{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}
Map tile multi-tiles	/collections/{collectionId}/map/{styleId}/tiles/{tileMatrixSetId}
Map tile multi-tiles collections <sup>1</sup>	/map/tiles/{tileMatrixSetId}
Images	
Image set	/collections/{collectionId}/images
Image	/collections/{collectionId}/images/{imageId}

<sup>1</sup>: In the first column of the table, the word "collections" means "from more than one collection"

# Chapter 7. Requirements Class "Images Core"

## 7.1. Overview

Requirements Class	
<a href="http://www.opengis.net/spec/ogcapi-images-1/1.0/req/core">http://www.opengis.net/spec/ogcapi-images-1/1.0/req/core</a>	
Target type	Web API
Dependency	<a href="#">RFC 2616 (HTTP/1.1)</a>
Dependency	<a href="#">RFC 2818 (HTTP over TLS)</a>
Dependency	<a href="#">RFC 3339 (Date and Time on the Internet: Timestamps)</a>
Dependency	<a href="#">RFC 8288 (Web Linking)</a>
Dependency	<a href="http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core">http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core</a>
Dependency	<a href="http://www.opengis.net/spec/ogcapi-common-1/1.0/req/collections">http://www.opengis.net/spec/ogcapi-common-1/1.0/req/collections</a>

An API that implements this conformance class provides access to image resources organized in image-sets of a [dataset](https://www.w3.org/TR/vocab-dcat/#class-dataset) [https://www.w3.org/TR/vocab-dcat/#class-dataset]. In other words, the API is a [distribution](https://www.w3.org/TR/vocab-dcat/#class-distribution) [https://www.w3.org/TR/vocab-dcat/#class-distribution] of that dataset. An implementation of the OGC API - Tiles draft specification, for example, would be another distribution.

The entry point is a [Landing page](#) (path `/`).

The [Landing page](#) provides links to

- the [API definition](#) (path `/api`, link relation `service`),
- the [Conformance declaration](#) (path `/conformance`, link relation `conformance`), and
- the [Collections](#) (path `/collections`, link relation `data`). In this draft specification a collection represents an image set composed by individual images.

The [API definition](#) describes the capabilities of the API instance, adopting elements from the OGC API - Common draft specification and adding capabilities that can be used by clients to retrieve images via the API or by development tools to support the implementation of API enabled servers and clients. Accessing the [API definition](#) using HTTP GET returns a description of the OGC API - Images instance.

The [Conformance declaration](#) states the requirements classes from existing standards or community specifications. These are identified by a URI that the API conforms to. Clients can, but are not required to, use this information. Accessing the Conformance declaration using HTTP GET returns the list of URIs of requirements classes implemented by the OGC API Images.

This draft specification assumes that data is organized into one or more collections. [Collections](#) provides information about and access to the collections. Each set of images is represented by a `collectionId`. The collection can be retrieved as images, tiles or other representations - for example, a coverage (that might provide access to the image set as a mosaic).

**NOTE**

This specification reserves the term "collection" for the concept defined in OWS common (OGC 06-121r9) that is closer to the old concept of "layer" in WMS. In the OGC API - Images draft specification, images are organized in image sets (that will not be called collections of images to avoid confusion). An image set is a collection from the OGC API - Commons point of view.

Accessing **Collections** using HTTP GET returns a response that contains at least the list of collections. Accessing **Collections/{collectionId}** using HTTP GET returns a description of a collection with an indication if the collection can be retrieved as individual images. For each **Collection**, a link to metadata about the collection is available (path **/collections/{collectionId}**) with key information about the collection. This information includes:

- A local identifier for the collection that is unique for the dataset;
- An optional title and description for the collection;
- An optional extent that can be used to provide an indication of the spatial and temporal extent of the collection - typically derived from the data;
- Links to representations of the collections, presented as a single "image".

The **Collection** resource is available at path **/collections/{collectionId}**, often with more details than included in the **Collections** response.

## 7.2. General

Requirement 1	<b>/req/images/core/api-common</b>
A	The OGC API SHALL comply with the requirements specified in the <a href="http://www.opengis.net/spec/OAPI_Common/1.0/req/core">http://www.opengis.net/spec/OAPI_Common/1.0/req/core</a> and <a href="http://www.opengis.net/spec/OAPI_Common/1.0/req/collections">http://www.opengis.net/spec/OAPI_Common/1.0/req/collections</a> Requirements Classes of the OGC API - Common version 1.0 Standard.

In practice, this means that the landing page and the conformance page follow OGC API - Common core and collections requirements This standard provides extra additions to the OGC API - Common requirements that are particular to images.

## 7.3. API landing page

The landing page provides links to start exploring the resources offered by the API. The landing page mainly consists of a list of links. The OGC Web Service (OWS) Common Implementation Standard (OGC 06-121r9) already requires some common links that are enough for this draft specification core.

### 7.3.1. Response

No variations of the landing page are required. With a **/collections** successful response, the list of



collectionId and links to the /collections/{collectionId} can be retrieved. With a /collections/{collectionId} successful response, and if the collection is available as images, the links to retrieve and manage images can be discovered. Note that other resource types may also be retrieved as collections.

## 7.4. Declaration of conformance classes

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

### 7.4.1. Response

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

Requirement 2	/req/images/core/conformance-success
A	The API conformance path SHALL advertise the images core conformance class as links to <a href="http://www.opengis.net/spec/ogcapi-images-1/1.0/req/core">http://www.opengis.net/spec/ogcapi-images-1/1.0/req/core</a> .

In the conformance page (commonly in JSON format) the links are following the link schema defined in the OGC API - Common draft specification. The following is an example fragment of the response of a pure OGC API - Images conformance information page. A server may include conformance to other standards adding new capabilities to the OGC API Images or adding support to other resource types specified in other OGC API standards.

*Example 2. Conformance Information Page minimum example*

```
{
  "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-images-1/1.0/req/core"
  ]
}
```

## 7.5. Collections

This draft specification includes dependencies on OGC API - Common collections. Collections are mandatory in the core of this draft specification because collections are the object that will be included in a tile.

Collections will enumerate the collectionId available in this API as well as basic information about

each collectionId: id, title, description, extent, CRS and links. This common response is considered enough for a general description of the collection.

<b>Requirement 3</b>	<b>/req/images/core/ic-md-collection-links</b>
A	For each collection included in the response, a <b>links</b> property of the collection SHALL include a link to the description of the individual collection (relation: <b>item</b> ) (in addition to other links specified in OGC API Commons).

More specific details about the collection can be found following the link to the individual collections that follow the pattern /collections/{collectionId}

**NOTE**

The collectionId can be interpreted as the name of the image set composed by all the images in the collection. It can be also interpreted as the mosaic that can be created by combining all the images in the collection.

## 7.6. Collection

This draft specification includes dependencies on an OGC API - Common structured collection. The response of the operation is extended with a new link for the images description.

### 7.6.1. Collection Links

<b>Requirement 4</b>	<b>/req/images/core/sic-images-example</b>
A	If the resource can be retrieved as images, a <b>200</b> -response SHALL include at least a link to the image set description for this resource.
B	These links SHALL provide a URL with the fragment <b>/images</b> .

## 7.7. Image set description

The response of the image set description operation provides metadata about the image set (description, extent, license, etc.) as well as the list of links to the individual images currently in the image set. The information of the image set is generated by the server based on internal state of the images. The information can only be retrieved but this core does not provide any mechanism to create, modify or delete it, except the update of the metadata produced as a result of adding individual images.

### 7.7.1. Operation

<b>Requirement 5</b>	<b>/req/images/core/sci-op</b>
----------------------	--------------------------------

A	Every resource available as images SHALL support an operation to retrieve the description of the image set, available as a HTTP GET request to a URI that is composed of two parts: An initial part is the URI of a resource that can be represented as images (e.g. a collection with id /collections/{collectionId}) and the second part follows the pattern /images. Only the resources or collection that supports this operation can be retrieved as images.
---	---

Typical resources that can be retrieved as images follow the pattern /collections/{collectionId}. In this case collectionId is expected as a parameter.

### 7.7.2. Parameter collectionId

If the resource is a collection, a collectionId parameter is expected. This parameter is described in the OGC API - Common draft specification.

### 7.7.3. Parameters limit and offset

These parameters provide a mechanism to limit the number of images linked to an image-set and to allow for paging the results in case there are too many images. These parameters are described in the OGC API - Common draft specification.

### 7.7.4. Parameters bbox and datetime

These parameters provide a mechanism to limit the bounding box and temporal extent of images linked to image-set. These parameters are described in the OGC API - Common.

### 7.7.5. Response

A successful response of an image-set request for a collection that can be retrieved as images is a data structure describing the image set and listing the available images.

<b>Requirement 6</b>	<b>/req/images/core/sci-stac-col</b>
A	The content of the response of a successful execution SHALL be a JSON file following the STAC collection specification and schema.
B	The <i>rel</i> value <i>item</i> shall be used to enumerate the image links in and image set
<b>Recommendation 1</b>	<b>/rec/images/core/sci-stac-col-link-image</b>

A	The link to an individual image SHOULD be composed of two parts: An initial part is the URI of a resource that can be represented as images (e.g. a collection with id /collections/{collectionId}) and the second part follows the pattern /images/{imageId}.
---	--

The STAC collection specification is still evolving. At the time of writing this draft specification, the version available is 0.8.0. That version satisfied Testbed-15 requirements. New versions may appear in the future. This draft specification does not specify any requirement about the version of the STAC collection that should be used.

For the purpose of facilitating the elaboration of OpenAPI document, the OpenAPI 3.0 version of the schema to validate the response should also be included as part of the OpenAPI document. A STAC schema adapted to the OpenAPI 3.0 schema is provided in the annex [STAC Schemas and examples](#)

The following response illustrates a valid STAC collection document version 0.8.0

*Example 3. Example of the response to an image set request that follows the STAC collection format*

```
{
  "stac_version": "0.8.0",
  "stac_extensions": [],
  "id": "Daraa_mosaic_2019",
  "title": "Imagery from Daraa 2019",
  "description": "Catalog of imagery in coverage Daraa_mosaic_2019",
  "links": [
    {
      "rel": "self",
      "href": "https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Daraa_mosaic_2019/images"
    },
    {
      "rel": "root",
      "href": "https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Daraa_mosaic_2019/images"
    },
    {
      "rel": "item",
      "href": "https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Daraa_mosaic_2019/images/23"
    },
    {
      "rel": "item",
      "href": "https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Daraa_mosaic_2019/images/5"
    }
  ],
  "summaries": {},
  "keywords": "Daraa",
}
```

```

"version": "1.0.0",
"license": "proprietary",
"providers": [
  {
    "name": "NGA",
    "description": "National Geospatial-Intelligence Agency",
    "roles": [
      "producer"
    ],
    "url": "https://www.nga.mil"
  }
],
"extent": {
  "spatial": {
    "bbox": [
      [
        35.99601745605469,
        32.50426061084722,
        36.08110063620433,
        32.72028872300039
      ]
    ]
  },
  "temporal": {
    "interval": [
      [
        "2019-05-21T19:36:44Z",
        "2019-06-12T20:32:21Z"
      ]
    ]
  }
},
"properties": {}
}

```

## 7.8. Image description

The response to an Image description operation provides metadata about the image (description, extent, license, etc.) as well as a list of links to the main file of the image and all additional files that may accompany the image. The information about the image is generated by the server based on internal metadata about the image that is stored by the server during the addition or modification of an image in the image set. A way to submit or alter the image description is not provided by this draft specification. The image can also be deleted from the image set.

### 7.8.1. Operation

Requirement 7	/req/images/core/scii-op
---------------	--------------------------

A	An image set SHALL support an operation to retrieve the description of an image that is part of an image set, available as a HTTP GET request to a URI provided as part of the links section present in the image set description.
---	--

Typical resources that can be retrieved as images follow the pattern `/collections/{collectionId}`. In this case, `collectionId` is expected as a parameter.

The link to an individual image is normally composed of two parts: The first part is the URI of a resource that can be represented as images (e.g. a collection with id `/collections/{collectionId}`) and the second part follows the pattern `/images/{imageId}`. In this case `imageId` is expected as a parameter.

### 7.8.2. Parameter `collectionId`

If the resource is a collection, a `collectionId` parameter is expected. This parameter is described in the OGC API - Common draft specification.

### 7.8.3. Parameter `imageId`

If the resource follows the URI template recommendations, an `imageId` parameter is expected. The value of this parameter is the final name in the image path.

### 7.8.4. Response

A successful response to an image request is a data structure describing the image and linking to it. Sometimes an image is formed by more than one file (in here we call them **parts** of the image) and they are listed as other files related with the image.

<b>Requirement 8</b>	<b><code>/req/images/core/scii-stac-item</code></b>
A	The content of the response of a successful execution SHALL be a JSON file following the extension of the STAC item specification for images (that extends the original STAC item schema).
B	The image parts (files) SHALL be listed in the <i>assets</i> property of the response document. The main file of the image SHALL be included as a valid URL in the <i>href</i> property of <i>main</i> object in the <i>assets</i> object. The URLs of the other parts of an image might be contained in the <i>href</i> property in other objects of the <i>assets</i> object.
<b>Recommendation 2</b>	<b><code>/rec/images/core/sci-stac-item</code></b>

A	The <i>properties</i> object for the stac-item (that is inherited from GeoJSON) SHOULD have two properties <i>bbox</i> and <i>crs</i> that contains the native bounding box in the native CRS that represents the DomainSet of the Rangeset of values in the image (Domainset and Rangeset are concepts precisely defined for regular grids in the OGC CIS standard).
B	The <i>properties</i> element for the stac-item (that is inherited from GeoJSON) SHOULD have a property <i>nominalResM</i> that contains the nominal resolution of the data (expressed in meters).
B	The <i>properties</i> element for the stac-item (that is inherited from GeoJSON) SHOULD have a property <i>cloudCover</i> that contains the approximated percentage of the data that is covered of shadowed by clouds.

As explained before for the STAC collection, the STAC item specification is also still evolving. This draft specification does not specify any requirement about the version of the STAC item that should be used. For the purpose of facilitating the elaboration of an OpenAPI document, the OpenAPI 3.0 version of the schema to validate the response should also be included as part of the OpenAPI document. The STAC item schema adapted to the OpenAPI 3.0 schema is provided in [STAC Schemas and examples](#):

The following response illustrates a valid STAC item document, based on version 0.8.0 of the STAC specification.

*Example of the response to a image request that follows the STAC item format*

```
{
  "type": "Feature",
  "geometry": {
    "type": "Polygon",
    "coordinates": [
      [
        [
          35.99601746,
          32.57551892
        ],
        [
          36.08110064,
          32.57551892
        ],
        [
          36.08110064,
          32.64790382
        ],
        [
          35.99601746,
```

```

        32.64790382
    ],
    [
        35.99601746,
        32.57551892
    ]
]
},
"properties": {
    "datetime": "2019-05-21T19:36:44Z",
    "title": "One image",
    "license": "proprietary",
    "providers": [
        {
            "name": "NGA",
            "description": "National Geospatial-Intelligence Agency",
            "roles": [
                "producer"
            ],
            "url": "https://www.nga.mil"
        }
    ],
    "created": "2019-06-21T19:36:44Z",
    "updated": "2019-08-21T19:36:44Z",
    "nativeBbox": {
        "bbox": [
            780349.63,
            6556104.14,
            803726.72,
            6582470.15
        ],
        "crs": "http://www.opengis.net/def/crs/EPSSG/0/3857"
    },
    "nominalResM": 0.5332085767116562,
    "cloudCover": 80.5
},
"id": "LC81530252014153LGN00",
"links": [
    {
        "rel": "self",
        "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00.json"
    },
    {
        "rel": "alternate",
        "href": "https://landsatonaws.com/L8/153/025/LC81530252014153LGN00",
        "type": "text/html"
    },
    {
        "rel": "root",

```



```

    "href": "http://landsat-pds.s3.amazonaws.com/L8/L1T-collection.json"
  }
],
"stac_version": "0.8.0",
"stac_extensions": [
  "string",
  "checksum"
],
"assets": {
  "thumbnail": {
    "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_thumb_large.jpg",
    "type": "image/jpeg"
  },
  "metadata": {
    "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_MTL.txt",
    "type": "mtl"
  },
  "B1": {
    "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B1.TIF",
    "type": "image/vnd.stac.geotiff",
    "eo:bands": [
      0
    ]
  },
  "B2": {
    "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B2.TIF",
    "type": "image/vnd.stac.geotiff",
    "eo:bands": [
      1
    ]
  },
  "B3": {
    "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B3.TIF",
    "type": "image/vnd.stac.geotiff",
    "eo:bands": [
      2
    ]
  },
  "B4": {
    "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B4.TIF",
    "type": "image/vnd.stac.geotiff",
    "eo:bands": [
      3
    ]
  }
]

```

```

},
"B5": {
  "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B5.TIF",
  "type": "image/vnd.stac.geotiff",
  "eo:bands": [
    4
  ]
},
"B6": {
  "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B6.TIF",
  "type": "image/vnd.stac.geotiff",
  "eo:bands": [
    5
  ]
},
"B7": {
  "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B7.TIF",
  "type": "image/vnd.stac.geotiff",
  "eo:bands": [
    6
  ]
},
"B8": {
  "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B8.TIF",
  "type": "image/vnd.stac.geotiff",
  "eo:bands": [
    7
  ]
},
"B9": {
  "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B9.TIF",
  "type": "image/vnd.stac.geotiff",
  "eo:bands": [
    8
  ]
},
"B10": {
  "href": "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B10.TIF",
  "type": "image/vnd.stac.geotiff",
  "eo:bands": [
    9
  ]
},
"B11": {
  "href": "http://landsat-

```

```
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B11.TIF",
  "type": "image/vnd.stac.geotiff",
  "eo:bands": [
    10
  ]
},
"collection": "Landsat8",
"bbox": [
  35.99601745605469,
  32.57551891751058,
  36.08110063620433,
  32.64790382025549
]
}
```

# Chapter 8. Requirements Class "Images Transactional"

## 8.1. Overview

Requirements Class	
<a href="http://www.opengis.net/spec/ogcapi-images-1/1.0/req/transactional">http://www.opengis.net/spec/ogcapi-images-1/1.0/req/transactional</a>	
Target type	Web API
Dependency	<a href="#">RFC 2616 (HTTP/1.1)</a>
Dependency	<a href="#">RFC 2818 (HTTP over TLS)</a>
Dependency	<a href="#">RFC 3339 (Date and Time on the Internet: Timestamps)</a>
Dependency	<a href="#">RFC 8288 (Web Linking)</a>
Dependency	<a href="http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core">http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core</a>
Dependency	<a href="http://www.opengis.net/spec/ogcapi-common-1/1.0/req/collections">http://www.opengis.net/spec/ogcapi-common-1/1.0/req/collections</a>
Dependency	<a href="http://www.opengis.net/spec/ogcapi-images-1/1.0/req/core">http://www.opengis.net/spec/ogcapi-images-1/1.0/req/core</a>

The following API definition extends the OGC API - Images core to support the creation, update and deletion of images from an image set.

## 8.2. API landing page

The landing page provides links to start exploring the resources offered by the API. The landing page mainly consists of a list of links. OGC API - Common specifies common links that are required for the API image core.

### 8.2.1. Response

There are no required variations to the landing page.

## 8.3. 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 server API must declare the requirements classe(s) it implements and conforms to.

### 8.3.1. Response

The conformance page mainly consists of a list of links. OGC API - Common already requires some links that are visible in the next example.

<b>Requirement 9</b>	<b><code>/req/images/transactional/conformance-success</code></b>
----------------------	---

A	The OGC API - Images conformance path SHALL advertise the images transactional conformance class as links to <a href="http://www.opengis.net/spec/ogcapi-images-1/1.0/req/transactional">http://www.opengis.net/spec/ogcapi-images-1/1.0/req/transactional</a> .
---	--

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

*Example 4. Conformance Information Page fragment*

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

## 8.4. Image upload

There are two ways to upload an image to an image set based on the use of the HTTP POST and PUT verbs. The second way is also useful to modify an existing image.

To determine if the user has the necessary privileges to add an image to an existing image-set or to modify an image that is already part of an image set, use the OPTIONS verb to check the operations available to the user. For example, to determine if an image can be added to an image-set, an OPTIONS request to the image set URI should be performed. To determine if an image can be modified, an OPTIONS request to the image URI should be performed.

### 8.4.1. Operation

Requirement 10	/req/images/transactional/sci-post-op
A	Every image URI SHALL support the HTTP OPTIONS operation
B	An image set that accepts the addition of new images SHALL advertise support of a POST operation as part of the headers ("Accept" header) of a successful response to a request OPTIONS operation to the URI of the image set.

C	If the OPTIONS operation to the image set includes POST, the image set URI SHALL accept a POST operation to new image to an image set.
D	As a result of a successful operation, the image SHALL become available as a new image of the image set.

Typical new resources can be uploaded as images following the pattern /collections/{collectionId}. In this case collectionId is expected as a parameter.

### 8.4.2. Parameter collectionId

If the resource is a collection, a collectionId parameter is expected. This parameter is described in the draft OGC API – Common specification.

### 8.4.3. Payload

<b>Requirement 11</b>	<b>/req/images/transactional/sci-post-payload</b>
A	The payload sent in the body of the HTTP POST request SHALL be a new image compatible with the image set. The selected media type SHALL be capable of containing the image values as well as all mandatory metadata in the STAC item schema.

Examples of valid media types are a geo-enabled standalone file (e.g. image/tiff; application=geotiff.) or a multipart format that includes a part with value and auxiliary parts with the metadata (application/vnd.ogc.multipart;container=application/x-zip-compressed;images=image/tiff).

### 8.4.4. Response

A successful response to an image upload will return an HTTP Status Code 201 and will contain the URL to retrieve the new image created.

<b>Requirement 12</b>	<b>/req/images/transactional/sci-uri</b>
A	The Location header in the response SHALL contain the URI to the new image created in the server..

#### NOTE

The HTTP status response codes are specified in the HTTP standard. The specific recommended usage in the OGC APIs can be found in the OGC API - Common draft specification.

## 8.4.5. Operation

Requirement 13	<b>/req/images/transactional/scii-put-op</b>
A	Every image URI SHALL support the HTTP OPTIONS operation.
B	An image that can be modified in the server SHALL advertise support to a PUT operation as part of the headers ("Accept" header) of a successful response to a request OPTIONS operation to the URI of the image.
C	If the OPTIONS operation to the image includes PUT, the image URI SHALL accept a PUT operation to a new or modified image in the image set.
D	If the image is new, it SHALL become available as a new image of the image set. If the image was already in the image set, the server SHALL delete the old image and replace it with the new image.

Typical resources can be uploaded as images in image-sets that follow the pattern `/collections/{collectionId}`. In this case `collectionId` is expected as a parameter.

The link to an individual image is normally comprised of two parts: An initial part that is the URI of a resource that can be represented as images (e.g. a collection with id `/collections/{collectionId}`) and the second part that follows the pattern `/images/{imageId}`. In this case `imageId` is expected as a parameter.

## 8.4.6. Parameter `collectionId`

If the resource is a collection, a `collectionId` parameter is expected. This parameter is described in OGC API - Common.

## 8.4.7. Parameter `imageId`

If the resource follows the URI template recommendations, an `imageId` parameter is expected. The value of this parameter is the final name in the image path.

## 8.4.8. Request payload

Requirement 14	<b>/req/images/transactional/scii-put-payload</b>
A	The payload sent in the body of the HTTP PUT request SHALL be a new image compatible with the image set. The selected media type SHALL be capable of containing the image values as well as all mandatory metadata in the STAC item schema.

Examples of valid media types are a geo-enabled standalone file (e.g. image/tiff; application=geotiff.) or a multipart format that includes a part with value and auxiliary parts with the metadata (application/vnd.ogc.multipart;container=application/x-zip-compressed;images=image/tiff).

### 8.4.9. Response

A successful response to an image upload will return a HTTP Status Code 201 if the resource does not exist or an HTTP Status Code 200 if the resource has been modified.

### 8.4.10. Operation

Requirement 15	/req/images/transactional/scii-delete-op
A	An image that can be deleted from the server SHALL advertise support for a DELETE operation as part of the headers ("Accept" header) of a successful response to a request OPTIONS operation to the URI of the image.
B	If the OPTIONS operation to the image includes DELETE, the image URI SHALL accept a DELETE operation to an existing image in the image set.
C	As a result of a successful operation, the server SHALL remove the image from the image-set.

Typical resources that can be retrieved as images follow the pattern /collections/{collectionId}. In this case, collectionId is expected as a parameter.

The link to an individual image is normally comprised of two parts: The first part is the URI of a resource that can be represented as images (e.g. a collection with id /collections/{collectionId}) and the second part follows the pattern /images/{imageId}. In this case, imageId is expected as a parameter.

### 8.4.11. Parameter collectionId

If the resource is a collection, a collectionId parameter is expected. This parameter is described in OGC API - Common.

### 8.4.12. Parameter imageId

If the resource follows the URI template recommendations, an imageId parameter is expected. The value of this parameter is the final name in the image path.

### 8.4.13. Response

A successful response to an image deletion will return a HTTP Status Code 200



# Chapter 9. Requirement Class "Changeset core"

## 9.1. Overview

Requirements Class	
<a href="http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/core">http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/core</a>	
Target type	Web API
Dependency	<a href="#">RFC 2616 (HTTP/1.1)</a>
Dependency	<a href="#">RFC 2818 (HTTP over TLS)</a>
Dependency	<a href="#">RFC 3339 (Date and Time on the Internet: Timestamps)</a>
Dependency	<a href="#">RFC 8288 (Web Linking)</a>
Dependency	<a href="http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core">http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core</a>

The "Changeset core" is an extension building block to the OGC API that is applicable to any data resource that adds a mechanism to request from a server changes in a resource collection since a previous checkpoint. There is no HTTP verb that allows for a partial retrieval of a collection resource, so this capability is implemented as a GET request but is in fact as a partial GET.

At some point in time, the client will request a resource collection and receive that collection with a checkpoint identifier. At a later date and time, the client may send a request for updates and notify the server about the original checkpoint associated with the client status (that is short identifier for the resources the client already knows). The server will then inform the client about the changes in the resources up to now and communicate the current checkpoint. The client can then apply the update and subsequently update its local copy of the resources. Checkpoint identifiers should be considered as unsorted strings and the client should not make any assumption on the order of Ids based on the characters present (despite the fact that some servers could implement checkpoint identifiers as incremental numbers for convenience). This specification does not provide any mechanism to recover a previous status from the server; The changeset mechanism only provides the update from a known checkpoint to the present server status.

Checkpoints should be used with care if a complete synchronization of changes between client and server is required. For example, if the client requests changes using a checkpoint with a spatial filter or a priority filter, the client receives a new checkpoint that is only useful with a follow up request that uses the same filter. If the client intends to request changes using a less restrictive filter, the client should use an older checkpoint to be sure that no changes are lost.

While dealing with atomic resources, consider reusing e-tags to detect changes on cached versions of them as defined in <https://tools.ietf.org/html/rfc7232>. The e-tag mechanism permits the client to provide a fast, unmodified response for the case in which the resource has not changed. If the resource has been modified, that resource is completely downloaded. The e-tag mechanism works well in "all-or-nothing" situations, but it is not suitable for a collection resource when only some elements of the collection are updated. This requirement class specifies the capability for partial updates to a collection and the server will only retrieve the resource completely if all its parts have

changed.

## 9.2. General

<b>Requirement 16</b>	<b>/req/changeset/core/api-common</b>
A	The OGC API SHALL comply with the requirements specified in the <a href="http://www.opengis.net/spec/OAPI_Common/1.0/req/core">http://www.opengis.net/spec/OAPI_Common/1.0/req/core</a> and collections Requirements Classes of the OGC API-Common version 1.0 Standard.

In practice, the above requirement dictates that the landing page and the conformance page follow the OGC API - Common requirements. This draft specification defines extensions to the OGC API - Common requirements that are particular to changesets.

## 9.3. API landing page

The changeset API has been designed using query parameters that can potentially be added to any relevant URL. The landing page provides links to start exploring the resources offered by the API but does not expose checkpoints. The landing page mainly consists of a list of links. OGC API - Common already requires a fix set of common links that are enough for this core (see the OGC API - Common for details).

### 9.3.1. Response

There are no required variations to the landing page.

## 9.4. Declaration of conformance classes

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

### 9.4.1. Response

The conformance page mainly consists of a list of links. OGC API - Common specifies required links for any OGC API instance.

<b>Requirement 17</b>	<b>/req/changeset/core/conformance-success</b>
A	The API conformance path SHALL advertise the changeset core conformance class as links to <a href="http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/core">http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/core</a> .

On the conformance page (commonly in JSON format), the links follow the link schema defined in

OGC API – Common. The following is an example fragment of the response of an implementation of the OGC API – Map draft specification and an OGC API - Tiles draft specification (also developed in Testbed-15) conformance information page with changeset support.

*Example 5. 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/styles",
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core",
    "http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/core",
  ]
}
```

## 9.5. Resource retrieval

### 9.5.1. Headers

A header in the response of a resource retrieval is added to a successful GET request with the checkpoint. This header will be added to a successful response of complete GET request of with a checkpoint parameter.

Requirement 18	/req/changeset/core/resource-success
A	The body of a successful response SHALL include a header x-checkpoint with the value of the current server checkpoint.

This is necessary for the client to be able to formulate a changeset update request later.

## 9.6. Changeset extra parameters

A resource retrieval GET request can have additional query parameters to specify a filter to get only the changes in a resource collection from the last client request. The client saves checkpoint information associated with the resources it has saved or cached and the client can communicate this information to the server in a follow up request. The server will be able to know what elements the client has and deliver only the updated elements (and the deleted elements).

For a low connectivity environment, a parameter is used to select the priority level of the update changes and request only the most significant changes.

An additional parameter allows selecting the amount of content and the format of the response.

### 9.6.1. Parameter Checkpoint

Requirement 19	/req/changeset/core/cp-checkpoint-definition
A	<p>The operation SHALL support an optional parameter <code>checkpoint</code> with the following characteristics (using an OpenAPI Specification 3.0 fragment)</p> <pre>name: checkPoint in: query required: false schema:   type: string</pre>
B	<p>If the parameter is not specified in the request, the response SHALL contain all changes since auditing began.</p>

### 9.6.2. Parameter priority

Requirement 20	/req/changeset/core/cp-priority-definition
A	<p>The operation SHALL support an optional parameter <code>priority</code> with the following characteristics (using an OpenAPI Specification 3.0 fragment)</p> <pre>name: priority in: query required: false style: form explode: false schema:   type: string   default: all   enum:     - low     - medium     - high     - all   example: high</pre>
B	<p>If the parameter is not specified in the request, the response SHALL contain all changes whatever their priority.</p>

### 9.6.3. Parameter changeSetType

Requirement 21	/req/changeset/core/cp-changesettype-definition
A	<p>The operation SHALL support an optional parameter <code>changeSetType</code> that determines the type of the response and with the following characteristics (using an OpenAPI Specification 3.0 fragment) and determines the type of the response</p> <pre data-bbox="438 472 1323 1037">name: changeSetType in: query required: false style: form explode: false schema:   type: string   default: full   enum:     - summary     - full     - package example: full</pre>
B	<p>If the value of the <code>changeSetType</code> parameter is set to <code>full</code> or if the parameter is not specified in the request, the server SHALL return the complete <code>changeSet</code> file (with the changes embedded in the document) following the <code>changeSet</code> schema.</p>
C	<p>If the value of the <code>changeSetType</code> parameter is set to <code>summary</code> the server SHALL return a document with the summary of the changes. The summary SHALL follow the <code>changeSetSummary</code> schema. The server will not return the actual representation of the changes.</p>
D	<p>If the value of the <code>changeSetType</code> parameter is set to <code>package</code> the server SHALL return a package (e.g. a ZIP file) that will include a document with the summary of the changes (the summary follows the <code>changeSetSummary</code> schema) (with no actual changes embedded) and the changes themselves as separated parts in the package. Use this value for requesting changes for tiled resources.</p>

## 9.7. Changeset response

### 9.7.1. Changeset response with changes

A successful response to a resource request that has the checkpoint query filter will change to a format that contains a changeSet report and eventually the relevant changes.

Requirement 22	/req/changeset/core/cp-success
A	A successful execution of the operation that has found changes SHALL be reported as a response with a HTTP status code 200.
B	If the <b>checkPoint</b> property is present in the response, it SHALL contain the checkpoint value from which new changes are requested (not the new checkpoint).
C	The response SHALL contain a property <b>summaryOfChangedItems</b> containing a count of changes corresponding to each specific priority label.
D	If the <b>extentOfChangedItems</b> property is present in the response, it SHALL contain a list with one or more of bounding boxes affected by changes produced and in which CRS the coordinates of the bounding boxes are produced.
E	If the <b>numberOfReturnedItems</b> property is present in the response, it SHALL contain the number of changed items that are presented in the response.
F	If the <b>changedItems</b> property is present in the response, it SHALL contain a list with the new or modified resources (embedded or linked).
G	If the <b>deletedItems</b> property is present in the response, it SHALL contain a list of identifiers of deleted resources.

H

If a summary of changes has been requested, the content of that response SHALL contain a changeSetSummary report based upon the following OpenAPI 3.0 schema:

```
title: changeSetSummary
description: A document containing a summary of the
updates since a specified checkpoint
type: object
required:
- summaryOfChangedItems
properties:
  summaryOfChangedItems:
    description: a per-priority list of change counts
    type: array
    items:
      description: a count of changes corresponding to
a specific priority label
      type: object
      required:
      - priority
      - count
      properties:
        priority:
          description: the priority label
          type: string
        count:
          description: the count of changes tagged
with the corresponding priority
          label
          type: integer
      example:
      - priority: high
        count: 2
      - priority: medium
        count: 4
      - priority: low
        count: 67
```

I

If a changes report has been requested, the content of that response SHALL contain a changeSet report based upon the following OpenAPI 3.0 schema:

```
title: changeSet
description: A document containing the updates since a
specified checkpoint
allOf:
- $ref: '#/components/schemas/changeSetSummary'
- type: object
  properties:
    checkPoint:
      description: the checkpoint value from which new
changes are requested. This checkpoint is part of the
request to get this file.
      type: string
      example: "cp143756"
    extentOfChangedItems:
      description: extent of the new, changed or
deleted items
      type: object
      required: bbox
      properties:
        bbox:
          type: array
          items:
            $ref:
'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/bbox'
        crs:
          $ref:
'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/crs'
    numberOfReturnedItems:
      description: the number of changed items that
are presented in the response; this may be less than the
total number of changes
      type: integer
      example: 6
    changedItems:
      description: a list of new or modified
resources. Useful for features. Not recommended for
tiles
      type: array
      items:
        description: a representations of or reference
to the changed resource; e.g.
a GeoJSON-encoded feature
        type: object
        required:
```



## 9.7.2. Changeset response with no changes

Requirement 23	/req/changeset/tiles/cp-not-modified
A	A successful execution of the operation that has NOT found changes SHALL be reported as a response with a HTTP status code 304.

```

    - type: object
    - $ref:
      'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
      common/1.0.0#/components/schemas/link'
    example:
      - priority: high
        items: []
      - priority: medium
        items: []
    deletedItems:
      description: a list of identifiers of deleted
        resources. Useful for features. Not recommended for
        tiles.
      type: array
      items:
        description: the identifier of a deleted
        feature
        type: object
        required:
          - items
        properties:
          priority:
            description: a priority label
            type: string
          items:
            type: array
            items:
              type: string
    example:
      - priority: medium
        items:
          - "/collections/BUILDINGS/items/F4674"
          - "/collections/BUILDINGS/items/F37465819"
  
```

# Chapter 10. Requirement Class "Changeset tiles"

## 10.1. Overview

<b>Requirements Class</b>	
<a href="http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/tiles">http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/tiles</a>	
Target type	Web API
Dependency	<a href="http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core">http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core</a>
Dependency	<a href="http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/core">http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/core</a>

This extension describes specific requirements for tiles in a changeSet. This class extends:

- "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/multitiles",
- "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/cols-multitiles" and
- "http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/core".

## 10.2. API landing page

There are no required variations to the landing page.

## 10.3. Declaration of conformance classes

### 10.3.1. Response

The conformance page mainly consists of a list of links. OGC API - Common specifies required links.

<b>Requirement 24</b>	<b><a href="/req/checkpoint/tiles/conformance-success">/req/checkpoint/tiles/conformance-success</a></b>
A	The API conformance path SHALL advertise the capability of generating tiles from multiple collections adding the conformance class for this capability as a link to <a href="http://www.opengis.net/spec/ogcapi-checkpoint-1/1.0/req/tiles">http://www.opengis.net/spec/ogcapi-checkpoint-1/1.0/req/tiles</a> .

On the conformance page (commonly in JSON format), the links follow the link schema defined in OGC API – Common. The following is an example fragment of the response to an OGC API -Tiles draft specification (also developed in Testbed-15) conformance information page request:

```
{
  "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/styles",
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core",
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/tmxs",
    "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/multitiles",
    "http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/core",
    "http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/tiles"
  ]
}
```

## 10.4. Multi-tiles operation returning a changeset

A changeset is encoded as a list of tiles that have been affected by changes. The OGC API – Tiles draft specification provides a mechanism to request more than one tile in a single operation. The result is a multi-tile format that warps the tiles in a single package. The same mechanism can be adapted to return one the tiles affected by changes since the last checkpoint.

The multi-tiles operation is applied on top of the requirements class "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/multitiles" or "http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/cols-multitiles". The parameters defined in "http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/core" are added to the parameters defined for multi-tiles or cols-multitiles (multi-tiles of tiles from multiple collections). For requesting changesets of tiles there are some restrictions to the parameters defined in the OGC API - Changeset core that are described in this section.

For completeness, relevant parameters in the OGC API - Tiles multi-tiles are reviewed here.

### 10.4.1. Parameter priority

This parameter is defined in the OGC API - Changeset core. The use of priority makes sense for feature-based data where changes in some elements such as main roads or bridges may be considered more important than paths or buildings and therefore have high priority for an update. For tiles, this parameter usage is less obvious whether there is applicability. There is still a use case for imagery-based map tiles where this parameter could be of use. In a set of images that have been photo-interpreted, there could be annotations that are attached to the imagery that reveal places that are considered of higher importance (e.g. detection of strategic infrastructures or military arsenals) that then have a higher priority.

The meaning of the priority levels for tiles is left undefined in this draft specification and the parameter can be ignored by clients and services.

## 10.4.2. Parameter multiTileType

This parameter is defined in OGC API - Tiles multitiles. Of the three possible values defined in the original draft specification, only *full* is relevant here.

<b>Requirement 25</b>	<b>/req/changeset/tiles/REQ_cp-multitiletype</b>
A	Clients and services SHALL ignore the parameter <i>multiTileType</i> and behave as if the value of the parameter is <i>full</i> .

## 10.4.3. Parameter changeSetType

<b>Requirement 26</b>	<b>/req/changeset/tiles/cp-changesettiles</b>
A	Clients and services SHALL ignore the parameter <i>changeSetType</i> and behave as if the value of the parameter is <i>package</i> .

Note that the multi-tile operation with the Checkpoint parameter has a different output than the "pure" multi-tile operation. In practice, this means that the parameter *changeSetType* overwrites the parameter *multiTileType* and *changeSetType* takes precedence.

## 10.5. Changeset response

The Changeset report is the one defined in the core but with the addition of an extra element to document the scale range affected by the changes.

<b>Requirement 27</b>	<b>/req/checkpoint/tiles/cp-success</b>
A	If the changes in the tiles only affect a range of scales, a <i>scalesOfChangedItems</i> property SHALL be present in the changeset in the request indicating the interval of scales affected by changes. If this parameter is not included, all scales that are affected by changes.
B	If the <i>extentOfChangedItems</i> property is present in the response the CRS of the coordinates SHALL be the same as the <i>TileMatrixSet</i> .

C	<p>When the <code>changeSetType</code> parameter forces a response using the <code>changeSet</code> schema, the content of that response SHALL use the <code>changeSetTiles</code> schema instead (that extends <code>changeSet</code> by adding a scale interval). The <code>changeSetTiles</code> OpenAPI 3.0 follows the schema:</p> <pre> allOf: - \$ref: '#/components/schemas/changeSet' - type: object   properties:     scalesOfChangedItems:       type: object       required:         - minScaleDenominator:         - maxScaleDenominator:       properties:         minScaleDenominator:           type: number           format: double         maxScaleDenominator:           type: number           format: double </pre>
D	<p>When a package is being returned and the package format supports expressing file paths of its parts (such as a ZIP file), each tile in the package SHALL have a path following the template: <code>{tileMatrixSetId}/{tileMatrix}/{tileRow}/{tileCol}.{file-extension}</code>. <code>{file-extension}</code> is the file extension that corresponds to the media type (e.g "jpg" for image/jpeg).</p>

# Appendix A: Abstract Test Suite

An Abstract Test Suite is specified in Clause 9 and Annex A of ISO 19105. That Clause and Annex specify the ISO/TC 211 requirements for Abstract Test Suites.

**NOTE**

This section is intentionally not provided in this Engineering Report. It should be expanded by a SWG.

# Appendix B: STAC Schemas and examples

The STAC specification is still evolving. At the time of writing this standard, the version available was 0.8.0 and it satisfied the Testbed-15 requirements. New versions may appear in the future. This draft specification does not specify any requirement about the version of the STAC that should be used.

## B.1. STAC collection schema

For the purpose of facilitating the elaboration of OpenAPI document, the OpenAPI 3.0 version of the schema to validate the response should also be included as part of the OpenAPI document. The original schema can be found in <https://github.com/radiantearth/stac-spec/tree/master/collection-spec>. The STAC collection schema adapted to the OpenAPI 3.0 schema is provided here:

```
title: STAC Collection Specification
description:
  This object represents Collections in a SpatioTemporal Asset Catalog. Stac
  Collection is described in https://github.com/radiantearth/stac-
  spec/blob/master/collection-spec/collection-spec.md
  This schema is a replica of https://raw.githubusercontent.com/radiantearth/stac-
  spec/master/collection-spec/json-schema/collection.json
type: object
allof:
  - $ref: '#/components/schemas/stacCatalogue'
  - title: STAC Collection
    description: These are the fields specific to a STAC Collection. All other
    fields
      are inherited from STAC Catalog.
type: object
required:
  - license
  - extent
properties:
  stac_extensions:
    title: STAC extensions
    type: array
    uniqueItems: true
    items:
      anyOf:
        - title: Reference to a JSON Schema
          type: string
          format: uri
        - title: Reference to a core extension
          type: string
          enum:
            - asset
            - checksum
            - scientific
example: []
```

```
keywords:
  title: Keywords
  type: array
  items:
    type: string
  example: Daraa
version:
  title: Collection Version
  type: string
  example: 1.0.0
license:
  title: Collection License Name
  type: string
  example: proprietary
providers:
  $ref: '#/components/schemas/stacProviders'
extent:
  title: Extents
  type: object
  required:
    - spatial
    - temporal
  properties:
    spatial:
      title: Spatial extent object
      type: object
      required:
        - bbox
      properties:
        bbox:
          title: Spatial extents
          type: array
          minItems: 1
          items:
            title: Spatial extent
            type: array
            minItems: 4
            maxItems: 6
            items:
              type: number
      example:
        bbox:
          - 35.99601745605469
          - 32.50426061084722
          - 36.08110063620433
          - 32.72028872300039
    temporal:
      title: Temporal extent object
      type: object
      required:
        - interval
```



```

    properties:
      interval:
        title: Temporal extents
        type: array
        minItems: 1
        items:
          title: Temporal extent
          type: array
          minItems: 2
          maxItems: 2
          items:
            type:
              - string
              - 'null'
            format: date-time
      example:
        interval:
          - '2019-05-21T19:36:44Z'
          - '2019-06-12T20:32:21Z'
    properties:
      title: Common properties
      type: object
  stacCatalogue:
    title: STAC Catalog Specification
    description:
      This object represents Catalogs in a SpatioTemporal Asset Catalog.
      This schema is deeply inspired on
      https://raw.githubusercontent.com/radiantearth/stac-spec/master/catalog-spec/json-schema/catalog.json
      Changes
      The link definition has been replaced by the one in OGC API Common.
    type: object
    required:
      - stac_version
      - id
      - description
      - links
    properties:
      stac_version:
        title: STAC version
        type: string
        const: 0.8.0
        example: 0.8.0
      stac_extensions:
        title: STAC extensions
        type: array
        uniqueItems: true
        items:
          type: string
    id:
      title: Identifier

```

```

    type: string
    example: Daraq_mosaic_2019
  title:
    title: Title
    type: string
    example: Imagery from Daraq 2019
  description:
    title: Description
    type: string
    example: Catalog of imagery in coverage Daraq_mosaic_2019
  links:
    title: Links
    type: array
    items:
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/Link'
    example:
      - rel: self
        href: https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Daraq_mosaic_2019/images
      - rel: root
        href: https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Daraq_mosaic_2019/images
      - rel: item
        href: https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Daraq_mosaic_2019/images/23
      - rel: item
        href: https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Daraq_mosaic_2019/images/5
    summaries:
      $ref: '#/components/schemas/stacSummaries'
  stacSummaries:
    type: object
  additionalProperties:
    oneOf:
      - title: Stats
        type: object
        required:
          - min
          - max
        properties:
          min:
            title: Minimum value
            type:
              - number
              - string
          max:
            title: Maximum value
            type:
              - number
              - string

```

```

- title: Set of values
  type: array
  minItems: 1
  items:
    description: Any data type could occur.
stacProviders:
  type: array
  items:
    properties:
      name:
        title: Organization name
        type: string
        example: NGA
      description:
        title: Provider description
        type: string
        example: National Geospatial-Intelligence Agency
      roles:
        title: Organization roles
        type: array
        items:
          type: string
          enum:
            - producer
            - licensor
            - processor
            - host
      url:
        title: Homepage
        type: string
        format: url
        example: https://www.nga.mil

```

## B.2. STAC item schema extended for images

For the purpose of facilitating the elaboration of an OpenAPI document, the OpenAPI 3.0 version of the schema to validate the response should also be included as part of the OpenAPI document. The original schema can be found in <https://github.com/radiantearth/stac-spec/tree/master/item-spec>. The STAC item schema adapted to the OpenAPI 3.0 schema is provided below in the object stacItem. This scheme also includes a stacItemImage object that extends the stacItem schema with specific elements defined in this OGC API Images draft specification.

```

stacItemImage:
  description: OGC API Image extension for STAC Item Adding some properties specific
  for imagery. (The proposed element `citation` has not been included because there is a
  properties/providers that can do the job partially)
  allOf:
    - $ref: '#/components/schemas/stacItem'
    - type: object

```

```

properties:
  geometry:
    type: object
    properties:
      type:
        example: Polygon
      coordinates:
        type: array
        items:
          type:
            - number
            - array
        example:
          - - - 35.99601746
            - 32.57551892
          - - 36.08110064
            - 32.57551892
          - - 36.08110064
            - 32.64790382
          - - 35.99601746
            - 32.64790382
          - - 35.99601746
            - 32.57551892
  bbox:
    type: array
    items:
      type: number
    example:
      - 35.99601745605469
      - 32.57551891751058
      - 36.08110063620433
      - 32.64790382025549
  properties:
    type: object
    properties:
      nativeBbox:
        description: Native bounding box of the image.
        type: object
        required: bbox
        properties:
          bbox:
            description: |-
              One bounding boxes that describe the spatial extent of the
              dataset.
              West, south, east, north edges of the bounding box. The
              coordinates are in the coordinate reference system specified in `crs`. By default this
              is WGS 84 longitude/latitude.
            type: array
            minItems: 4
            maxItems: 4
            items:

```

```

        type: number
        example:
          - 780349.63
          - 6556104.14
          - 803726.72
          - 6582470.15
        crs:
          description: |-
            Coordinate reference system of the coordinates in the `bbox`. The
            default reference system is WGS 84 longitude/latitude.
          type: string
          default: 'http://www.opengis.net/def/crs/OGC/1.3/CRS84'
          example: 'http://www.opengis.net/def/crs/EPSSG/0/3857'
        nominalResM:
          description: Nominal resolution (in meters) of the image
          type: number
          format: double
          example: 0.5332085767116562
        cloudCover:
          description: Percentage of cloud cover
          type: number
          format: double
          example: 80.5
    stacItem:
      title: STAC Item
      description:
        This object represents the metadata for an item in a SpatioTemporal Asset
        Catalog.
        This schema is deeply inspired on https://github.com/radianteearth/stac-spec/blob/master/item-spec/item-spec.md
        Changes
        The link definition has been replaced by the one in OGC API Common.
        The https://geojson.org/schema/Feature.json has been replaced by the GeoJSON
        geometry description in OGC API features.
      type: object
      allOf:
        - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-features/1.0.0#/components/schemas/featureGeoJSON'
        - type: object
      required:
        - stac_version
        - id
        - geometry
        - links
        - assets
        - bbox
        - properties
      properties:
        stac_version:
          title: STAC version
          type: string

```

```

const: 0.8.0
example: 0.8.0
stac_extensions:
  title: STAC extensions
  type: array
  uniqueItems: true
  items:
    anyOf:
      - title: Reference to a JSON Schema
        type: string
        format: uri
      - title: Reference to a core extension
        type: string
        enum:
          - checksum
          - cube
          - datetime-range
          - eo
          - label
          - pointcloud
          - sar
          - scientific
id:
  title: Provider ID
  description: Provider item ID
  type: string
  example: LC81530252014153LGN00
links:
  title: Item links
  description: Links to item relations
  type: array
  items:
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/link'
  example:
    - rel: self
      href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00.json"
    - rel: alternate
      href: "https://landsatonaws.com/L8/153/025/LC81530252014153LGN00"
      type: text/html
    - rel: root
      href: "http://landsat-pds.s3.amazonaws.com/L8/L1T-collection.json"
assets:
  title: Asset links
  description: Links to assets
  type: object
  #Next code has been commented to prevent a Error in the Swagger HUB
  #patternProperties:
  #  ".+":
  #    $ref: '#/components/schemas/stacAsset'

```

```
additionalProperties: false
example:
  thumbnail:
    href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_thumb_large.jpg"
    type: image/jpeg
  metadata:
    href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_MTL.txt"
    type: mtl
  B1:
    href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B1.TIF"
    type: image/vnd.stac.geotiff
    eo:bands:
      - 0
  B2:
    href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B2.TIF"
    type: image/vnd.stac.geotiff
    eo:bands:
      - 1
  B3:
    href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B3.TIF"
    type: image/vnd.stac.geotiff
    eo:bands:
      - 2
  B4:
    href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B4.TIF"
    type: image/vnd.stac.geotiff
    eo:bands:
      - 3
  B5:
    href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B5.TIF"
    type: image/vnd.stac.geotiff
    eo:bands:
      - 4
  B6:
    href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B6.TIF"
    type: image/vnd.stac.geotiff
    eo:bands:
      - 5
  B7:
    href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B7.TIF"
    type: image/vnd.stac.geotiff
```

```

    eo:bands:
      - 6
    B8:
      href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B8.TIF"
      type: image/vnd.stac.geotiff
      eo:bands:
        - 7
    B9:
      href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B9.TIF"
      type: image/vnd.stac.geotiff
      eo:bands:
        - 8
    B10:
      href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B10.TIF"
      type: image/vnd.stac.geotiff
      eo:bands:
        - 9
    B11:
      href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B11.TIF"
      type: image/vnd.stac.geotiff
      eo:bands:
        - 10
  properties:
    type: object
    required:
      - datetime
    properties:
      datetime:
        title: Date and Time
        description: 'The searchable date/time of the assets, in UTC (Formatted
in
          RFC 3339) '
        type: string
        format: date-
        example: '2019-05-21T19:36:44Z'
      title:
        title: Item Title
        description: A human-readable title describing the item.
        type: string
        example: One image
      license:
        title: Item Licenses
        type: string
        example: proprietary
    providers:
      $ref: '#/components/schemas/stacProviders'
  created:

```



```

    title: Metadata updated at
    type: string
    format: date-time
    example: '2019-06-21T19:36:44Z'
  updated:
    title: Metadata updated at
    type: string
    format: date-time
    example: '2019-08-21T19:36:44Z'
  collection:
    title: Collection ID
    description: The ID of the STAC Collection this Item references to.
    type: string
    example: Landsat8
  stacProviders:
    type: array
    items:
      properties:
        name:
          title: Organization name
          type: string
          example: NGA
        description:
          title: Provider description
          type: string
          example: National Geospatial-Intelligence Agency
        roles:
          title: Organization roles
          type: array
          items:
            type: string
            enum:
              - producer
              - licensor
              - processor
              - host
      url:
        title: Homepage
        type: string
        format: url
        example: https://www.nga.mil

```

This schema is dependent on the schema definition of featureGeoJSON that is included in the OGC API - Features Part 1: Core standard.

# Appendix C: OpenAPI Domain

## C.1. OpenAPI Domain images

This is the OGC API - Images domain OpenAPI developed and used in Testbed-15. This might be used and continued in a SWG in the future. This is based on and extracted from OpenAPI examples from the OGC API - Features - Part 1: Core standard (OGC 17-069r3).

This corresponds to a URL `.../ogc-api-images/1.0.0`

```
openapi: 3.0.2

info:
  description: STAC Domain
  version: '1.0.0'
  title: STAC Domain

components:
  schemas:
    stacItemImage:
      description: OGC API Image extension for STAC Item Adding some properties
        specific for imagery. (The proposed element `citation` has not been included because
        there is a properties/providers that can do the job partially)
      allOf:
        - $ref: '#/components/schemas/stacItem'
        - type: object
      properties:
        geometry:
          type: object
          properties:
            type:
              example: Polygon
            coordinates:
              type: array
              items:
                type:
                  - number
                  - array
          example:
            - - - 35.99601746
              - 32.57551892
            - - 36.08110064
              - 32.57551892
            - - 36.08110064
              - 32.64790382
            - - 35.99601746
              - 32.64790382
            - - 35.99601746
              - 32.57551892
```

```

bbox:
  type: array
  items:
    type: number
  example:
    - 35.99601745605469
    - 32.57551891751058
    - 36.08110063620433
    - 32.64790382025549
properties:
  type: object
  properties:
    nativeBbox:
      description: Native bounding box of the image.
      type: object
      required: bbox
      properties:
        bbox:
          description: |-
            One bounding boxes that describe the spatial extent of the

```

dataset.

West, south, east, north edges of the bounding box. The coordinates are in the coordinate reference system specified in `crs`. By default this is WGS 84 longitude/latitude.

```

      type: array
      minItems: 4
      maxItems: 4
      items:
        type: number
      example:
        - 780349.63
        - 6556104.14
        - 803726.72
        - 6582470.15

```

```

    crs:
      description: |-
        Coordinate reference system of the coordinates in the `bbox`.
        The default reference system is WGS 84 longitude/latitude.

```

```

      type: string
      default: 'http://www.opengis.net/def/crs/OGC/1.3/CRS84'
      example: 'http://www.opengis.net/def/crs/EPSG/0/3857'

```

```

    nominalResM:
      description: Nominal resolution (in meters) of the image
      type: number
      format: double
      example: 0.5332085767116562

```

```

    cloudCover:
      description: Percentage of cloud cover
      type: number
      format: double
      example: 80.5

```

```

stacItem:
  title: STAC Item
  description:
    This object represents the metadata for an item in a SpatioTemporal Asset
    Catalog.
    This schema is deeply inspired on https://github.com/radiantearth/stac-
    spec/blob/master/item-spec/item-spec.md
    Changes
    The link definition has been replaced by the one in OGC API Common.
    The https://geojson.org/schema/Feature.json has been replaced by the GeoJSON
    geometry description in OGC API features.
  type: object
  allOf:
    - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
    features/1.0.0#/components/schemas/featureGeoJSON'
    - type: object
      required:
        - stac_version
        - id
        - geometry
        - links
        - assets
        - bbox
        - properties
      properties:
        stac_version:
          title: STAC version
          type: string
          const: 0.8.0
          example: 0.8.0
        stac_extensions:
          title: STAC extensions
          type: array
          uniqueItems: true
          items:
            anyOf:
              - title: Reference to a JSON Schema
                type: string
                format: uri
              - title: Reference to a core extension
                type: string
                enum:
                  - checksum
                  - cube
                  - datetime-range
                  - eo
                  - label
                  - pointcloud
                  - sar
                  - scientific
  id:

```

```

title: Provider ID
description: Provider item ID
type: string
example: LC81530252014153LGN00
links:
  title: Item links
  description: Links to item relations
  type: array
  items:
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/link'
  example:
    - rel: self
      href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00.json"
    - rel: alternate
      href: "https://landsatonaws.com/L8/153/025/LC81530252014153LGN00"
      type: text/html
    - rel: root
      href: "http://landsat-pds.s3.amazonaws.com/L8/L1T-collection.json"
assets:
  title: Asset links
  description: Links to assets
  type: object
  #Next code has been commented to prevent a Error in the Swagger HUB
  #patternProperties:
  #  ".+":
  #    $ref: '#/components/schemas/stacAsset'
  additionalProperties: false
  example:
    thumbnail:
      href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_thumb_larg
e.jpg"
      type: image/jpeg
      metadata:
        href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_MTL.txt"
        type: mtl
      B1:
        href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B1.TIF"
        type: image/vnd.stac.geotiff
        eo:bands:
          - 0
      B2:
        href: "http://landsat-
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00_B2.TIF"
        type: image/vnd.stac.geotiff
        eo:bands:
          - 1

```

B3:

href: "http://landsat-  
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00\_B3.TIF"  
type: image/vnd.stac.geotiff  
eo:bands:  
- 2

B4:

href: "http://landsat-  
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00\_B4.TIF"  
type: image/vnd.stac.geotiff  
eo:bands:  
- 3

B5:

href: "http://landsat-  
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00\_B5.TIF"  
type: image/vnd.stac.geotiff  
eo:bands:  
- 4

B6:

href: "http://landsat-  
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00\_B6.TIF"  
type: image/vnd.stac.geotiff  
eo:bands:  
- 5

B7:

href: "http://landsat-  
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00\_B7.TIF"  
type: image/vnd.stac.geotiff  
eo:bands:  
- 6

B8:

href: "http://landsat-  
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00\_B8.TIF"  
type: image/vnd.stac.geotiff  
eo:bands:  
- 7

B9:

href: "http://landsat-  
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00\_B9.TIF"  
type: image/vnd.stac.geotiff  
eo:bands:  
- 8

B10:

href: "http://landsat-  
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00\_B10.TIF"  
type: image/vnd.stac.geotiff  
eo:bands:  
- 9

B11:

href: "http://landsat-  
pds.s3.amazonaws.com/L8/153/025/LC81530252014153LGN00/LC81530252014153LGN00\_B11.TIF"

```

    type: image/vnd.stac.geotiff
    eo:bands:
      - 10
  properties:
    type: object
    required:
      - datetime
    properties:
      datetime:
        title: Date and Time
        description: 'The searchable date/time of the assets, in UTC
          (Formatted in RFC 3339) '
        type: string
        format: date-
        example: '2019-05-21T19:36:44Z'
      title:
        title: Item Title
        description: A human-readable title describing the item.
        type: string
        example: One image
      license:
        title: Item Licenses
        type: string
        example: proprietary
      providers:
        $ref: '#/components/schemas/stacProviders'
      created:
        title: Metadata updated at
        type: string
        format: date-time
        example: '2019-06-21T19:36:44Z'
      updated:
        title: Metadata updated at
        type: string
        format: date-time
        example: '2019-08-21T19:36:44Z'
    collection:
      title: Collection ID
      description: The ID of the STAC Collection this Item references to.
      type: string
      example: Landsat8
  stacCollection:
    title: STAC Collection Specification
    description:

```

This object represents Collections in a SpatioTemporal Asset Catalog. Stac Collection is described in <https://github.com/radiantearth/stac-spec/blob/master/collection-spec/collection-spec.md>

This schema is a replica of <https://raw.githubusercontent.com/radiantearth/stac-spec/master/collection-spec/json-schema/collection.json>

```

type: object
allOf:
  - $ref: '#/components/schemas/stacCatalogue'
  - title: STAC Collection
    description: These are the fields specific to a STAC Collection. All other
fields
    are inherited from STAC Catalog.
type: object
required:
  - license
  - extent
properties:
  stac_extensions:
    title: STAC extensions
    type: array
    uniqueItems: true
    items:
      anyOf:
        - title: Reference to a JSON Schema
          type: string
          format: uri
        - title: Reference to a core extension
          type: string
          enum:
            - asset
            - checksum
            - scientific
      example: []
  keywords:
    title: Keywords
    type: array
    items:
      type: string
    example: Daraa
  version:
    title: Collection Version
    type: string
    example: 1.0.0
  license:
    title: Collection License Name
    type: string
    example: proprietary
  providers:
    $ref: '#/components/schemas/stacProviders'
  extent:
    title: Extents
    type: object
    required:
      - spatial
      - temporal
    properties:

```



```

spatial:
  title: Spatial extent object
  type: object
  required:
  - bbox
  properties:
    bbox:
      title: Spatial extents
      type: array
      minItems: 1
      items:
        title: Spatial extent
        type: array
        minItems: 4
        maxItems: 6
        items:
          type: number
  example:
    bbox:
      - - 35.99601745605469
      - 32.50426061084722
      - 36.08110063620433
      - 32.72028872300039
temporal:
  title: Temporal extent object
  type: object
  required:
  - interval
  properties:
    interval:
      title: Temporal extents
      type: array
      minItems: 1
      items:
        title: Temporal extent
        type: array
        minItems: 2
        maxItems: 2
        items:
          type:
          - string
          - 'null'
          format: date-time
  example:
    interval:
      - - '2019-05-21T19:36:44Z'
      - '2019-06-12T20:32:21Z'
  properties:
    title: Common properties
    type: object
stacCatalogue:

```

```

title: STAC Catalog Specification
description:
  This object represents Catalogs in a SpatioTemporal Asset Catalog.
  This schema is deeply inspired on
https://raw.githubusercontent.com/radiantearth/stac-spec/master/catalog-spec/json-
schema/catalog.json
  Changes
  The link definition has been replaced by the one in OGC API Common.
type: object
required:
- stac_version
- id
- description
- links
properties:
  stac_version:
    title: STAC version
    type: string
    const: 0.8.0
    example: 0.8.0
  stac_extensions:
    title: STAC extensions
    type: array
    uniqueItems: true
    items:
      type: string
  id:
    title: Identifier
    type: string
    example: Dareaa_mosaic_2019
  title:
    title: Title
    type: string
    example: Imagery from Dareaa 2019
  description:
    title: Description
    type: string
    example: Catalog of imagery in coverage Dareaa_mosaic_2019
  links:
    title: Links
    type: array
    items:
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/link'
    example:
      - rel: self
        href: https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Dareaa_mosaic_2019/images
      - rel: root
        href: https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Dareaa_mosaic_2019/images

```

```

    - rel: item
      href: https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Daraa_mosaic_2019/images/23
    - rel: item
      href: https://tb15.cubewerx.com/cubewerx/cubeserv/op-
stac/wmts/1.0.0/collections/Daraa_mosaic_2019/images/5
  summaries:
    $ref: '#/components/schemas/stacSummaries'
stacSummaries:
  type: object
  additionalProperties:
    oneOf:
      - title: Stats
        type: object
        required:
          - min
          - max
        properties:
          min:
            title: Minimum value
            type:
              - number
              - string
          max:
            title: Maximum value
            type:
              - number
              - string
      - title: Set of values
        type: array
        minItems: 1
        items:
          description: Any data type could occur.
stacAsset:
  type: object
  required:
    - href
  properties:
    href:
      title: Asset reference
      type: string
    title:
      title: Asset title
      type: string
    type:
      title: Asset type
      type: string
stacProviders:
  type: array
  items:
    properties:

```

```

name:
  title: Organization name
  type: string
  example: NGA
description:
  title: Provider description
  type: string
  example: National Geospatial-Intelligence Agency
roles:
  title: Organization roles
  type: array
  items:
    type: string
    enum:
      - producer
      - licensor
      - processor
      - host
url:
  title: Homepage
  type: string
  format: url
  example: https://www.nga.mil
parameters:
  imageId:
    name: imageId
    in: path
    description: local identifier of a image
    required: true
    schema:
      type: string

```

## C.2. OpenAPI Domain changset

This is the OGC API - Changeset domain OpenAPI developed and used in Testbed-15. This might be used and continued in a SWG in the future.

This corresponds to a URL `.../ogc-api-changeset/1.0.0`

```

openapi: 3.0.2
# Revisions:
# 2019-08-02 initial version

info:
  title: OGC API Checkpoint
  description: 'A delta update is the subset of resources that have changed between
two checkpoints and as such is itself a subset of resources from the specified
collection.
    Extracted from: https://portal.opengeospatial.org/wiki/Testbed15/DeltaUpdateApi'

```

```
version: '1.0.0'
```

```
components:
```

```
  parameters:
```

```
    checkPoint:
```

```
      name: checkPoint
```

```
      in: path
```

```
      description: Last checkpoint the client has retrieved data. Without it, the response to accessing the delta update path would be all changes since auditing began.
```

```
      required: false
```

```
      schema:
```

```
        type: string
```

```
    checkPoint-query:
```

```
      name: checkPoint
```

```
      in: query
```

```
      description: Last checkpoint the client has retrieved data. Without it, the response to accessing the delta update path would be all changes since auditing began.
```

```
      required: false
```

```
      schema:
```

```
        type: string
```

```
    changeSetType:
```

```
      name: changeSetType
```

```
      in: query
```

```
      description: |-
```

```
        'When successful, the service will respond to a query in one of two ways. It may either generate a complete response document containing resources that satisfy the operation or it may simply generate an empty response container that indicates the count of the total number of resources that the operation would return. Which of these two responses is generated is determined by the value of the optional changeSetType parameter.'
```

```
        The allowed values for this parameter are `full`, `summary` and `package`.
```

```
        If the value of the `changeSetType` parameter is set to `full` causes the server to return the complete changeSet file following the `changeSet` schema. Use this value to retrieve new features in the OGC API features as part of the change set file.
```

```
        If the value of the `changeSetType` parameter is set to `summary` causes the server to return a summary of the changes. The summary follows the `changeSetSummary` schema.
```

```
        If the value of the `changeSetType` parameter is set to `package` causes the server to return a package (e.g. a ZIP file) that will include a summary of the changes (The summary follows the `changeSetSummary` schema) and the changes themselves as separated parts in the package. Use this value for requesting changes in tiled resources
```

```
        The default value is `full`.
```

```
        NOTE in the original Peter Vretanos design this parameter was called `resultType` but it has been changed here to avoid conflicts with https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-common/1.0.0#/components/parameters/resultType
```

```
      required: false
```

```
      style: form
```

```

explode: false
schema:
  type: string
  default: full
  enum:
    - summary
    - full
    - package
example: full
priority:
  name: priority
  in: query
  description: |-
    One or more priority labels used to filter a delta update response.

    The allowed values for this parameter are 'low', 'medium', 'high' and 'all'

    The default value is 'all' for retrieve all changes.
  required: false
  style: form
  explode: false
  schema:
    type: string
    default: all
    enum:
      - low
      - medium
      - high
      - all
  example: high
schemas:
  changeSetSummary:
    title: changeSetSummary
    description: A document containing a summary of the updates since a specified
checkpoint
    type: object
    required:
      - summaryOfChangedItems
    properties:
      summaryOfChangedItems:
        description: a per-priority list of change counts
        type: array
        items:
          description: a count of changes corresponding to a specific priority
label
          type: object
          required:
            - priority
            - count
          properties:
            priority:

```

```

    description: the priority label
    type: string
  count:
    description: the count of changes tagged with the corresponding
priority
    label
    type: integer
  example:
    - priority: high
      count: 2
    - priority: medium
      count: 4
    - priority: low
      count: 67
  changeSet:
    title: changeSet
    description: A document containing the updates since a specified checkpoint
  allOf:
    - $ref: '#/components/schemas/changeSetSummary'
    - type: object
  properties:
    checkPoint:
      description: the checkpoint value from which new changes are requested.
This checkpoint is part of the request to get this file.
      type: string
      example: "cp143756"
    extentOfChangedItems:
      description: extents of the new, changed or deleted items
      type: array
      items:
        description: extent of the new, changed or deleted items
        type: object
        required: bbox
        properties:
          bbox:
            type: array
            items:
              $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/bbox'
          crs:
            $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/crs'
        numberOfReturnedItems:
          description: the number of changed items that are presented in the
response; this may be less than the total number of changes
          type: integer
          example: 6
        changedItems:
          description: a list of new or modified resources. Useful for features.
Not recommended for tiles
          type: array

```

```

    items:
      description: a representations of or reference to the changed
resource; e.g.
      a GeoJSON-encoded feature
      type: object
      required:
      - items
      properties:
        priority:
          description: a priority label
          type: string
        items:
          type: array
          items:
            oneOf:
              - type: object
              - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/Link'
        example:
          - priority: high
            items: []
          - priority: medium
            items: []
      deletedItems:
        description: a list of identifiers of deleted resources. Useful for
features. Not recommended for tiles.
        type: array
        items:
          description: the identifier of a deleted feature
          type: object
          required:
          - items
          properties:
            priority:
              description: a priority label
              type: string
            items:
              type: array
              items:
                type: string
          example:
            - priority: medium
              items:
                - "/collections/BUILDINGS/items/F4674"
                - "/collections/BUILDINGS/items/F37465819"
      changeSetTiles:
        title: changeSetTiles
        description: A document containing the tiles updated since a specified
checkpoint
        allOf:
        - $ref: '#/components/schemas/changeSet'

```



```
- type: object
  properties:
    scalesOfChangedItems:
      type: object
      description: Range of scales affected by the changes. If not present,
all scales are affected
      required:
        - minScaleDenominator:
        - maxScaleDenominator:
      properties:
        minScaleDenominator:
          type: number
          format: double
          example: 10000
        maxScaleDenominator:
          type: number
          format: double
          example: 10000000
```

# Appendix D: OpenAPI Examples

## D.1. OpenAPI Images example

In this example we present an imaginary API server that provides access to images using the OGC API - Images draft specification.

```
openapi: 3.0.1

servers:
  # Added by API Auto Mocking Plugin
  - description: SwaggerHub API Auto Mocking
    url: https://virtserver.swaggerhub.com/UAB-CREAF/ogc-api-images-opf-xml/1.0.0
  - description: Server
    url: http://data.example.org
info:
  version: '1.0.0'
  title: |-
    'Open Portrayal Framework Image API'
  description: 'This is a draft of the OGC Testbed-15 Image API in the Open
    Portrayal Framework task. The Image API is a Web API for fetching and managing a
    server that maintains products composed by lists of images. The typical example is a
    remote sensing product exposed a single mosaik but internally composed by many scenes
    that are constantly received from the satellite.'
  contact:
    name: Joan Masó
    email: joan.maso@uab.cat
  license:
    name: OGC License
    url: 'https://raw.githubusercontent.com/opengeospatial/WFS_FES/master/LICENSE'
tags:
  - name: OGC API Common
    description: |-
      Common characteristics of this API
  - name: Images
    description: |-
      access to Images
  - name: Manage Images
    description: |-
      Creation, modification of deletion of images
paths:
  '/':
    get:
      tags:
        - OGC API Common
      summary: landing page
      description: |-
        The landing page provides links to the API definition, the conformance
        statements and to the feature collections in this dataset.
```

```

operationId: getLandingPage
parameters:
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/f-json-html'
responses:
  '200':
    description: |-
      Links to the API capabilities shared by this API.
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/landingPage'
      text/html:
        schema:
          type: string
  '500':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
'/conformance':
get:
  summary: information about specifications that this API conforms to
  description: |-
    A list of all requirements classes specified in a standard that the
    server conforms to.
  operationId: getConformanceDeclaration
  parameters:
    - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/f-json'
  tags:
    - OGC API Common
  responses:
    '200':
      description: |-
        the URIs of all requirements classes supported by
        this API
      content:
        application/json:
          schema:
            $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/confClasses'
          example:
            - '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-images-1/1.0/req/core'
            - 'http://www.opengis.net/spec/ogcapi-images-
1/1.0/req/transactional'
    '400':
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/InvalidParam'
    '406':
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-

```

```

common/1.0.0#/components/responses/UnsupportedFormat'
  '500':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
'/collections':
  get:
    tags:
      - OGC API Common
    summary: the collections in the dataset
    operationId: getCollections
    parameters:
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/f-json-html'
    responses:
      '200':
        $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/Collections'
      '500':
        $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
'/collections/{collectionId}':
  get:
    tags:
      - OGC API Common
    summary: describe a collection
    operationId: describeCollection
    parameters:
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/collectionId'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/f-json-html'
    responses:
      '200':
        description: |-
          Metadata about the collection including image list
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/collection'
          text/html:
            schema:
              type: string
      '404':
        $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotFound'
      '500':
        $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
'/collections/{collectionId}/images':
  get:
    tags:

```

```

- Images
summary: retrieve images of the collection
operationId: getImages
parameters:
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/collectionId'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/limit'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/offset'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/bbox'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/datetime'
  - $ref: '#/components/parameters/f-stac-html'
responses:
  '200':
    description: |-
      Information about the feature collection plus the first features
      matching the selection parameters.
    content:
      application/stac+json:
        schema:
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
images/1.0.0#/components/schemas/stacCollection'
        text/html:
          schema:
            type: string
  '404':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotFound'
  '500':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
options:
  tags:
    - Manage images
summary: |-
  Options for creating new images
operationId: testSupportManageImages
description: |-
  Determines if POST is supported by this resource and user. See Accept headers
  for that.
parameters:
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/collectionId'
security:
  - BasicAuth: []
responses:
  '204':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-

```

```

common/1.0.0#/components/responses/NoContent'
  '401':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/UnauthorizedAccess'
post:
  tags:
    - Manage images
  summary: |-
    adds a new image
  operationId: addImage
  description: |-
    Adds an image to the image collection
    The URI of the new image is returned in the header `Location`.
    This operation is only available to registered image authors.
  security:
    - BasicAuth: []
  parameters:
    - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/collectionId'
  responses:
    '201':
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/Created'
    '400':
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/Invalid'
    '401':
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/UnauthorizedAccess'
  requestBody:
    description: |-
      image to be added in the form of a STAC document
    content:
      image/geo+tiff:
        schema:
          type: string
          format: binary
      application/vnd.ogc.multipart;container=application/x-zip-
compressed;images=geo+tiff:
        schema:
          type: string
          format: binary
      application/vnd.ogc.multipart;container=application/x-zip-
compressed;images=geo+tiff;description=application/stac+json:
        schema:
          type: string
          format: binary
  '/collections/{collectionId}/images/{imageId}':
get:
  tags:
    - Images

```

```

summary: fetch a image
operationId: getImage
parameters:
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/collectionId'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
images/1.0.0#/components/parameters/imageId'
  - $ref: '#/components/parameters/f-stac-html'
responses:
  '200':
    description: |-
      An image.
    content:
      application/stac+json:
        schema:
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
images/1.0.0#/components/schemas/stacItemImage'
      text/html:
        schema:
          type: string
  '404':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotFound'
  '500':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
options:
tags:
  - Manage images
summary: |-
  Options for modifying or deleting an image.
operationId: testSupportManageImage
description: |-
  Determines if PUTS or DELETE is suported by this resource and user. See
Allow headers for that.
parameters:
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/collectionId'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
images/1.0.0#/components/parameters/imageId'
security:
  - BasicAuth: []
responses:
  '204':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NoContent'
  '401':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/UnauthorizedAccess'
put:
tags:

```

- Manage images

summary: |-

replace an image resource or add a new one

description: |-

Replace an existing image with the id `imageId`. If no such image exists, a new resource with that id is added.

The resource will only be available in the native format in which the resource is posted. There is no support for automated conversions to other representations.

This operation is only available to registered image authors.

operationId: updateImage

security:

- BasicAuth: []

parameters:

- \$ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-common/1.0.0#/components/parameters/collectionId'

- \$ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-images/1.0.0#/components/parameters/imageId'

requestBody:

description: |-

A single image resource.

content:

application/stac+json:

schema:

\$ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-images/1.0.0#/components/schemas/stacItem'

image/geo+tiff:

schema:

type: string

format: binary

application/vnd.ogc.multipart;container=application/x-zip-compressed;images=geo+tiff:

schema:

type: string

format: binary

application/vnd.ogc.multipart;container=application/x-zip-compressed;images=geo+tiff;description=application/stac+json:

schema:

type: string

format: binary

responses:

'204':

\$ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-common/1.0.0#/components/responses/Updated'

'401':

\$ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-common/1.0.0#/components/responses/UnauthorizedAccess'

delete:

tags:

- Manage images



```

summary: |-
  delete an image resource
description: |-
  Delete an existing image with the id `imageId`. If no
  such resource exists, an error is returned.
  This operation is only available to registered image authors.
operationId: deleteImage
security:
  - BasicAuth: []
parameters:
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/collectionId'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
images/1.0.0#/components/parameters/imageId'
responses:
  '204':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/Deleted'
  '401':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/UnauthorizedAccess'
  '404':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotFound'

components:
  schemas:
    landingPage:
      allOf:
        - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/landingPage'
        - $ref: '#/components/schemas/landingPage-link'
    landingPage-link:
      #This element is a duplicate of the one in OGC API common but it is enriched
      with more examples for maps and other resource types.
      type: object
      required:
        - links
      properties:
        links:
          type: array
          items:
            $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/link'
          example:
            - href: 'http://data.example.org/?f=json'
              rel: self
              type: application/json
              title: this document
            - href: 'http://data.example.org/?f=html'
              rel: alternate

```

```

    type: text/html
    title: this document in HTML
  - href: 'http://data.example.org/api?f=json'
    rel: service
    type: application/vnd.oai.openapi+json;version=3.0
    title: the API definition in OpenAPI 3.0 JSON
  - href: 'http://data.example.org/api?f=html'
    rel: service
    type: text/html
    title: the API definition in HTML
  - href: 'http://data.example.org/conformance?f=json'
    rel: conformance
    type: application/json
    title: the list of conformance classes implemented by this API
  - href: 'http://data.example.org/collections?f=json'
    rel: data
    type: application/json
    title: The feature collections in the dataset in JSON
  - href: 'http://data.example.org/collections?f=html'
    rel: data
    type: text/html
    title: The feature collections in the dataset in HTML
  - href: 'http://data.example.org/tileMatrixSet?f=json'
    rel: tileMatrixSets
    type: application/json
    title: the list of tileMatrixSets implemented by this API in JSON
  - href: 'http://data.example.org/tileMatrixSet?f=html'
    rel: tileMatrixSets
    type: text/html
    title: the list of tileMatrixSets implemented by this API in HTML
collection:
  allOf:
    - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/collection'
    - $ref: '#/components/schemas/collection-link'
  collection-link:
    #This element is a duplicate of the one in OGC API common but it is enriched
with more examples for maps and other resource types.
    type: object
    required:
      - links
    properties:
      links:
        type: array
        items:
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/link'
        example:
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-collection-describedBy'
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-

```

```

common/1.0.0#/components/examples/link-collection-license-html'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-collection-license-rdf'
  - href: 'http://data.example.com/collections/buildings/images'
    rel: image
    type: 'image/geotiff'
parameters:
  f-stac-html:
    name: f
    in: query
    description: |-
      The format of the response. If no value is provided, the standard http rules
      apply, i.e., the accept header is used to determine the format.

      Pre-defined values are "application/stac+json" and "text/html".
    required: false
    style: form
    explode: false
    schema:
      type: string
      enum:
        - application/stac+json
        - text/html
    example: application/stac+json

```

## D.2. OpenAPI Images example

In this example we present an imaginary API server that provides access to tiles with changeset support using the OGC API - Changeset draft specification.

```

openapi: 3.0.0
# Revisions:
# 2019-07-14 Reparated from a more complex example

servers:
# Added by API Auto Mocking Plugin
  - description: SwaggerHub API Auto Mocking
    url: https://virtserver.swaggerhub.com/UAB-CREAF/ogc-api-tiles-opf-xmp-
changeset/1.0.0
  - description: SwaggerHub API Auto Mocking
    url: https://virtserver.swaggerhub.com/UAB-CREAF/ogc-api-tiles-opf-xmp-
changeset/1.0.0
  - description: SwaggerHub API Auto Mocking
    url: https://virtserver.swaggerhub.com/UAB-CREAF/ogc-api-map-tiles-opf-xmp-mt-
more-1-collection/1.0.0
  - description: SwaggerHub API Auto Mocking
    url: https://virtserver.swaggerhub.com/joanma747/opf-map-tiles-api/1.0.0
info:
  title: 'Open Portrayal Framework ChangeSet API'

```

description: |-

This is a draft of an example of a service following the OGC API changeset for tiled maps and features

It is elaborated in the Testbed-15 Open Portrayal Framework in collaboration with the WMS.SWG. The Map Tile API is a Web API for fetching and managing maps and tiles.

version: "1.0.0"

contact:

name: Joan Maso

email: joan.maso@uab.cat

license:

name: OGC License

url: 'https://raw.githubusercontent.com/opengeospatial/OGC-API-Map-

Tiles/master/LICENSE'

tags:

- name: OGC API Common

description: |-

Common characteristics of this API

- name: Map tiles metadata

description: |-

Metadata about map tiles and tileMatrixSets

- name: MultiTiles

description: |-

MultiTiles with changeset

- name: Features

description: |-

Features with changeset

paths:

'/':

get:

tags:

- OGC API Common

summary: landing page

description: |-

The landing page provides links to the API definition, the conformance statements and to the feature collections in this dataset.

operationId: getLandingPage

parameters:

- \$ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-common/1.0.0#/components/parameters/f-json-html'

responses:

'200':

description: |-

Links to the API capabilities and the TileMatrixSets shared by this API.

content:

application/json:

schema:

\$ref: '#/components/schemas/landingPage'

```

    text/html:
      schema:
        type: string
      '500':
        $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
      '/conformance':
        get:
          description: |-
            A list of all requirements classes specified in a standard that the
            server conforms to.
          operationId: getConformanceDeclaration
          parameters:
            - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/f-json'
          tags:
            - OGC API Common
          responses:
            '200':
              description: |-
                the URIs of all requirements classes supported by this API
              content:
                application/json:
                  schema:
                    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/confClasses'
                  example:
                    conformsTo:
                      # OGC API Common core consists on the landingPage, conformance
                      - 'http://www.opengis.net/spec/ogcapi-common-1/1.0/req/core'
                      # OGC API Common collections consists adds the capability to have
collections
                      - 'http://www.opengis.net/spec/ogcapi-common-
1/1.0/req/collections'
                      # We need to be sure which ones are still valid when adopting
OpenAPI.
                      - 'http://www.opengis.net/spec/ogcapi-features-1/1.0/req/core'
                      - 'http://www.opengis.net/spec/ogcapi-features-1/1.0/req/oas30'
                      - 'http://www.opengis.net/spec/ogcapi-features-1/1.0/req/html'
                      - 'http://www.opengis.net/spec/ogcapi-features-1/1.0/req/geojson'
                      - 'http://www.opengis.net/spec/ogcapi-maps-1/1.0/req/core'
                      - 'http://www.opengis.net/spec/ogcapi-maps-1/1.0/req/styles'
                      - 'http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/core'
                      - 'http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/tmxs'
                      - 'http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/collections'
                      - 'http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/multitiles'
                      - 'http://www.opengis.net/spec/ogcapi-tiles-1/1.0/req/cols-
multitiles'
                      - 'http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/core'
                      - 'http://www.opengis.net/spec/ogcapi-changeset-
1/1.0/req/features'

```

```

      - 'http://www.opengis.net/spec/ogcapi-changeset-1/1.0/req/tiles'
    '400':
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/InvalidParam'
    '406':
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/UnsupportedFormat'
    '500':
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
  '/collections':
    get:
      tags:
        - OGC API Common
      summary: the collections in the dataset
      operationId: getCollections
      parameters:
        - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/f-json-html'
      responses:
        '200':
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/Collections'
        '500':
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
  '/collections/{collectionId}':
    get:
      tags:
        - OGC API Common
      summary: describe a collection
      operationId: describeCollection
      parameters:
        - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/collectionId'
        - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/f-json-html'
      responses:
        '200':
          description: |-
            Metadata about the collection including style information.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/collection'
            text/html:
              schema:
                type: string
        '404':
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotFound'

```

```

    '500':
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
  '/tileMatrixSets':
    get:
      tags:
        - Map tiles metadata
      summary: fetch all available tile matrix sets (tiling schemes)
      operationId: getTileMatrixSets
      parameters:
        - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/f-json'
      responses:
        '200':
          description: |-
            An array of tile matrix sets (tiling schemes).
          content:
            application/json:
              schema:
                $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/schemas/tileMatrixSets'
        '500':
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
  '/tileMatrixSets/{tileMatrixSetId}':
    get:
      tags:
        - Map tiles metadata
      summary: fetch a tile matrix sets (tiling scheme) by id
      operationId: getTileMatrixSetDescription
      parameters:
        - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/parameters/tileMatrixSetId'
        - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/f-json'
      responses:
        '200':
          description: tile matrix sets (a tiling scheme).
          content:
            application/json:
              schema:
                $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/schemas/tileMatrixSet'
        '404':
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotFound'
        '500':
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
  '/collections/{collectionId}/items':
    get:

```

```

tags:
  - OGC API Features
summary: retrieve features of the feature collection
description: |-
  IMPORTANT: This operation is defined by in OGC API features. It is included
  here for INFORMATION ONLY But is OUT OF SCOPE for Maps and Tiles.
operationId: getFeatures
parameters:
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
  common/1.0.0#/components/parameters/collectionId'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
  common/1.0.0#/components/parameters/limit'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
  common/1.0.0#/components/parameters/offset'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
  common/1.0.0#/components/parameters/bbox'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
  common/1.0.0#/components/parameters/datetime'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
  common/1.0.0#/components/parameters/resultType'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
  changeset/1.0.0#/components/parameters/checkPoint-query'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
  changeset/1.0.0#/components/parameters/priority'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
  features/1.0.0#/components/parameters/properties'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
  common/1.0.0#/components/parameters/f-json-html'
  - name: f_code
    in: query
    description: Filter the collection by f_code
    required: false
    style: form
    explode: false
    schema:
      type: string
responses:
  '200':
    description: |-
      Information about the feature collection plus the first features
      matching the selection parameters.
    content:
      application/geo+json:
        schema:
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
          features/1.0.0#/components/schemas/featureCollectionGeoJSON'
      application/changeset+json:
        schema:
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
          changeset/1.0.0#/components/schemas/changeSet'
      text/html:

```



```

        schema:
          type: string
      '304':
        $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotModified'
      '404':
        $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotFound'
      '500':
        $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
    '/collections/{collectionId}/map/tiles':
      get:
        tags:
          - Map tiles metadata
        summary: fetch a map tiles description
        description: |-
          Retrieves the tiles description for this collection including the `links` to
          get a `tile`, the `TileMatrixSetLink` and the `infoTemplate`
        operationId: describeMapTiles
        parameters:
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/collectionId'
        responses:
          '200':
            description: |-
              Description of the map tiles.
            content:
              application/json:
                schema:
                  $ref: '#/components/schemas/map-tiles'
          '404':
            $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotFound'
          '500':
            $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
    '/map/tiles':
      get:
        tags:
          - Map tiles metadata
        summary: fetch a map tiles description
        description: |-
          Retrieves the map tiles description for this collection including the
          `links` to get a `tile`
        operationId: describeMapTilesCollections
        responses:
          '200':
            description: |-
              Links to the map tiles.
            content:

```

```

application/json:
  schema:
    type: object
    required:
      - links
    properties:
      links:
        type: array
        items:
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/link'
        example:
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
map-tiles/1.0.0#/components/examples/link-map-tiles-col-this'
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
map-tiles/1.0.0#/components/examples/link-map-tiles-col-tile'
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
map-tiles/1.0.0#/components/examples/link-map-tiles-col-info'
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
map-tiles/1.0.0#/components/examples/link-map-tiles-col-multitile'
      '404':
        $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotFound'
      '500':
        $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
    '/collections/{collectionId}/map/{styleId}/tiles/{tileMatrixSetId}':
      get:
        tags:
          - MultiTiles
        summary: |-
          fetch tiles of a collection.
        operationId: getMapTilesCollectionId
        parameters:
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/collectionId'
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-map-
styles/1.0.0#/components/parameters/styleId'
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
maps/1.0.0#/components/parameters/transparent'
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
maps/1.0.0#/components/parameters/bgcolor'
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/parameters/tileMatrixSetId'
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/parameters/scaleDenominator'
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/bbox'
          - $ref: '#/components/parameters/f-tile'
          - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/parameters/multiTileType'

```

```

- $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
changeset/1.0.0#/components/parameters/checkPoint-query'
- $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
changeset/1.0.0#/components/parameters/changeSetType'
- $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
changeset/1.0.0#/components/parameters/priority'
- $ref: '#/components/parameters/f-json-zip'
responses:
  '200':
    description: |-
      A tiling scheme used to partition the collection into tiles.
    content:
      application/changeset+json:
        schema:
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
changeset/1.0.0#/components/schemas/changeSetTiles'
      application/json:
        schema:
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/schemas/tileSet'
      application/vnd.ogc.multipart;container=application/x-zip-
compressed;tiles=image/png:
        schema:
          type: string
          format: binary
  '304':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotModified'
  '404':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotFound'
  '500':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
'/map/tiles/{tileMatrixSetId}':
  get:
    tags:
      - MultiTiles
    summary: |-
      fetch tiles of multiple collections.
    operationId: getMapTilesCollections
    parameters:
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-map-
tiles/1.0.0#/components/parameters/collections'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-map-
styles/1.0.0#/components/parameters/styles'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
maps/1.0.0#/components/parameters/transparent'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
maps/1.0.0#/components/parameters/bgcolor'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-

```

```

tiles/1.0.0#/components/parameters/tileMatrixSetId'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/parameters/scaleDenominator'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/parameters/bbox'
  - $ref: '#/components/parameters/f-tile'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/parameters/multiTileType'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
changeset/1.0.0#/components/parameters/checkPoint-query'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
changeset/1.0.0#/components/parameters/changeSetType'
  - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
changeset/1.0.0#/components/parameters/priority'
  - $ref: '#/components/parameters/f-json-zip'
responses:
  '200':
    description: |-
      A tiling scheme used to partition the collection into tiles.
    content:
      application/changeset+json:
        schema:
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
changeset/1.0.0#/components/schemas/changeSetTiles'
      application/json:
        schema:
          $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/schemas/tileSet'
      application/vnd.ogc.multipart;container=application/x-zip-
compressed;tiles=image/png:
        schema:
          type: string
          format: binary
  '304':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotModified'
  '404':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/NotFound'
  '500':
    $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/responses/ServerError'
components:
  schemas:
    landingPage:
      allOf:
        - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/landingPage'
        - $ref: '#/components/schemas/landingPage-link'
    landingPage-link:
      #This element is a duplicate of the one in OGC API common but it is enriched

```

with more examples for maps and other resource types.

```
type: object
required:
  - links
properties:
  links:
    type: array
    items:
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/link'
    example:
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-landingPage-this'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-landingPage-alternate'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-landingPage-service-json'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-landingPage-service-html'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-landingPage-conformance'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-landingPage-collections-json'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-landingPage-collections-html'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/examples/link-landingPage-tms-json'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-map-
tiles/1.0.0#/components/examples/link-landingPage-map-tiles'
  collection:
    allOf:
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/collection'
      - $ref: '#/components/schemas/collection-link'
  collection-link:
```

#This element is a duplicate of the one in OGC API common but it is enriched with more examples for maps and other resource types.

```
type: object
required:
  - links
properties:
  links:
    type: array
    items:
      $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/link'
    example:
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-collection-this'
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-collection-describedBy'
```

```

- $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-collection-license-html'
- $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/examples/link-collection-license-rdf'
- $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
features/1.0.0#/components/examples/link-collection-items'
- $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-map-
tiles/1.0.0#/components/examples/link-collection-map-tiles'
  map-tiles:
    allOf:
      - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
tiles/1.0.0#/components/schemas/tiles'
      - $ref: '#/components/schemas/map-tiles-link'
    map-tiles-link:
      #This element is a duplicate of the one in OGC API common but it is enriched
with more examples for maps and other resource types.
      type: object
      required:
        - links
      properties:
        links:
          type: array
          items:
            $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-
common/1.0.0#/components/schemas/link'
          example:
            - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-map-
tiles/1.0.0#/components/examples/link-map-tiles-this'
            - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-map-
tiles/1.0.0#/components/examples/link-map-tiles-tile'
            - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-map-
tiles/1.0.0#/components/examples/link-map-tiles-info'
            - $ref: 'https://api.swaggerhub.com/domains/UAB-CREAF/ogc-api-map-
tiles/1.0.0#/components/examples/link-map-tiles-multitile'
      parameters:
        f-json-zip:
          name: f
          in: query
          description: |-
            The format of the response. If no value is provided, the standard http
rules apply, i.e., the accept header is used to determine the format.

            The only pre-defined value is "json". The response to other values is
determined by the server.
          required: false
          style: form
          explode: false
          schema:
            type: string
            enum:
              - json

```

```
- zip
example: json
f-png-jpeg:
name: f
in: query
description: |-
```

The format of the response. If no value is provided, the standard http rules apply, i.e., the accept header is used to determine the format.

Pre-defined values jpeg, png or gif for image based tiles

The response to other values is determined by the server.

```
required: false
style: form
explode: false
schema:
  type: string
  enum:
    - png
    - jpeg
    - gif
example: png
```

```
f-tile:
name: f-tile
in: query
description: |-
```

The format of the response. If no value is provided, the standard http rules apply, i.e., the accept header is used to determine the format.

Pre-defined values are jpeg, png or gif for image based tiles

The response to other values is determined by the server.

```
required: false
style: form
explode: false
schema:
  type: string
  enum:
    - image/png
    - image/jpeg
    - image/gif
example: image/png
```

# Appendix E: Revision History

Table 2. Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
August 28, 2019	J. Masó	-	all	initial version based on an implementation of Keith Pomakis
October 14, 2019	J. Masó	0.8	all	ready for OGC review
October 21, 2019	J. Masó	0.9	all	results of the OGC review
October 25, 2019	J. Masó	1.0	all	document ready for the 3 week rule in the Toulouse TC.
December 12, 2019	C. Reed	1.0	Various	Minor edits to Abstract, Exec Summary, and various other clauses prior to publication as a Public ER
January 3, 2020	G. Hobona	1.0	Various	Final OGC staff review and edits.



# Appendix F: Bibliography

1. STAC Community: SpatioTemporal Asset Catalog (STAC), <https://github.com/radiantearth/stac-spec>, (2019).
2. OpenAPI Initiative: The OpenAPI Specification, <https://github.com/OAI/OpenAPI-Specification>, (2019).