

OGC API – Maps Sprint 2020
Summary Engineering Report

Publication Date: 2021-01-18

Approval Date: 2020-12-15

Submission Date: 2020-11-16

Reference number of this document: OGC 20-090

Reference URL for this document: <http://www.opengis.net/doc/PER/OGCAPIMapsSprint1>

Category: OGC Public Engineering Report

Editor: Gobe Hobona

Title: OGC API – Maps Sprint 2020: Summary Engineering Report

OGC Public Engineering Report

COPYRIGHT

Copyright © 2021 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 NON-INFRINGEMENT 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.

Table of Contents

1. Subject	5
2. Executive Summary	6
2.1. Document contributor contact points	6
2.2. Foreword	7
3. References	8
4. Terms and definitions	9
4.1. Abbreviated terms	9
5. Introduction	10
5.1. Participants	10
5.2. OGC API – Maps	11
6. High-Level Architecture	12
7. Results	13
8. Discussion	17
8.1. Lessons Learned	19
9. Conclusions	20
9.1. Future work	20
9.1.1. Ideas for the Innovation Program	20
9.1.2. Ideas for the Standards Program	20
Appendix A: Revision History	21
Appendix B: Bibliography	22

Chapter 1. Subject

This OGC Engineering Report (ER) documents the results and recommendations from a code sprint that was held from 28 to 29 July 2020 to advance the development of the draft OGC API – Maps Standard. An Application Programming Interface (API) is a standard set of documented and supported functions and procedures that expose the capabilities or data of an operating system, application, or service to other applications (adapted from ISO/IEC TR 13066-2:2016). The OGC API - Maps Sprint was an online virtual event. The sprint was sponsored by Ordnance Survey.

Chapter 2. Executive Summary

This Engineering Report (ER) summarizes the main achievements of the OGC API – Maps Sprint. The sprint served to accelerate development of the draft OGC API – Maps Standard. The objectives of the code sprint were to:

- Develop prototype implementations of OGC API – Maps
- Test the prototype implementations
- Provide feedback to the Editor about what worked and what did not work
- Provide feedback about the specification document

Part of the motivation for holding the sprint was:

- The increasing need for interoperability between Web APIs
- The growing uptake of location information within non-traditional geospatial developer communities
- Maps continue to be key decision-making support tools

The draft OGC API - Maps standard describes an API that presents data as maps by applying a style. The draft standard enables a client application to request maps as images. This includes the ability to specify or change parameters such as the size of an image and coordinate reference systems at the time of request. The draft standard supports the creation of implementer-friendly API definitions that are easily understandable by developers with little or no geospatial experience. These maps can be retrieved as images of any size generated on-the-fly or in a tiled structure (if used alongside the draft OGC API - Tiles standard).

A server that implements the draft OGC API - Maps standard provides information about available maps, produces them, and answers queries regarding their content. The draft standard comes from the OGC's concerted effort to create modular, resource-oriented API standards that use OpenAPI for describing interfaces that offer geographic information over the web. These OGC APIs are collectively known as the OGC API family of draft and approved standards. The draft OGC API - Maps standard addresses use cases similar to those addressed by the OGC Web Map Service (WMS) standard.

The sprint produced very valuable results. Participants identified areas for improving the draft standard, allowing the editor of the draft standard to record their recommendations. In some cases, the specification editor was able to fix issues in real time as they those issues were being identified.

2.1. Document contributor contact points

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

Contacts

Name	Organization	Role
Gobe Hobona	OGC	Editor

2.2. 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 3. References

The following normative documents are referenced in this document.

- OGC: OGC 17-069r3, OGC API - Features - Part 1: Core 1.0 (2019)
- OGC: OGC 06-042, OpenGIS Web Map Service (WMS) Implementation Specification 1.3.0 (2006)
- OGC: OGC 05-078r4, Styled Layer Descriptor, Version 1.1 (2007)
- OGC: OGC 19-072, OGC API - Common - Part 1: Core candidate standard
- OGC: OGC 20-058, OGC API - Maps - Part 1: Core candidate standard
- OGC: OGC 20-057, OGC API - Tiles candidate standard
- IETF: RFC-7946 The GeoJSON Format (2016)

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.

- **coordinate reference system**

coordinate system that is related to the real world by a datum term name (source: ISO 19111)

- **OpenAPI Document**

A document (or set of documents) that defines or describes an API. An OpenAPI definition uses and conforms to the OpenAPI Specification (<https://www.openapis.org>)

- **portrayal**

presentation of information to humans (source: ISO 19117)

- **style**

a sequence of rules of symbolizing instructions to be applied by a rendering engine on one or more features and/or coverages

- **Web API**

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

NOTE

See Best Practice 24: Use Web Standards as the foundation of APIs (W3C Data on the Web Best Practices) for more detail.

4.1. Abbreviated terms

- **API** Application Programming Interface
- **CRS** Coordinate Reference System
- **OGC** Open Geospatial Consortium
- **WMS** Web Map Service

Chapter 5. Introduction

The code sprint was held to focus on the draft OGC API – Maps standard. Sprint participants prototyped implementations of the draft OGC API - Maps - Part 1: Core standard, identifying missing requirements, and documenting these requirements so that the draft standard could be improved.

An OGC Code Sprint is a collaborative and inclusive event driven by innovative and rapid programming with minimal process and organization constraints. OGC sprints are designed to support the development of new applications and open standards. OGC Code Sprints experiment with emerging ideas in the context of geospatial standards, help improve interoperability of existing standards by experimenting with new extensions or profiles, and are used as a proof of concept for other OGC Innovation Program initiatives, or support OGC Standards Program activities.

This code sprint was held as part of a series in the third quarter of 2020. The other code sprints included:

- OGC API – Coverages Sprint held August 10th to 11th, 2020. This sprint focused on the draft OGC API - Coverages - Part 1: Core Standard which defines a Web API for publishing and accessing coverage data.
- OGC API – Common and OGC API – Features Sprint held September 29th to 30th, 2020. This sprint included work on the draft OGC API - Common - Part 2: Collections Standard which provides a common connection between the API landing page and resource-specific details. The sprint also covered the draft OGC API – Features – Part 4: Simple Transactions Standard which extends the core capabilities specified in the OGC API – Features – Part 1: Core standard with the ability to add, replace, modify, and/or delete individual feature instances from a single feature collection.

5.1. Participants

Software developers and solutions architects from the following organizations registered to participate in the code sprint:

- Brad Hards (individual)
- Carter Jonas
- CREAf-UAB
- CubeWerx Inc.
- Ecere Corporation
- ECMWF
- Geobeyond
- Geosolutions Consulting
- Giscorps
- Global Nomad GIS Services

- Heazeltech
- Image Matters LLC
- ISTAR
- Land Information New Zealand
- Met Office
- Natural Resources Canada
- Ordnance Survey
- Polaris Digitech Limited
- Princeton University
- Technical University in Munich (TUM)
- Texas A&M University
- UAB-CREAF
- US Army Geospatial Center
- US Census
- Wuhan University

5.2. OGC API – Maps

The draft OGC API - Maps Standard describes an API that presents maps portraying data that has been rendered according to a style. The maps served by implementations of the draft OGC API - Maps Standard are retrieved as images of any size, generated on-the-fly, and with the styling determined by the client application. The draft standard can be considered the successor to the widely implemented WMS standard. Note that the WMS is known to be the OGC standard with the greatest number of implementations.

This draft standard covers the following conformance classes:

- The *Maps* core conformance class describes how to transform a data resource into a map resource in the default style.
- The *Maps styles* conformance class describes how to transform a data resource into a map resource applying a style.
- The *Map maps* conformance class proposes a mechanism to deliver maps. This mechanism requires further elaboration and discussion in the WMS Standards Working Group (SWG).
- The *Maps collections* conformance class describes how to transform a list of data resources (collections) into a map resource.

Chapter 6. High-Level Architecture

The focus of the sprint was on support of development of the draft OGC API – Map Standard. Implementations of this draft standard were deployed in each participant’s own infrastructure in order to build a solution with the architecture shown below in [Figure 1](#).

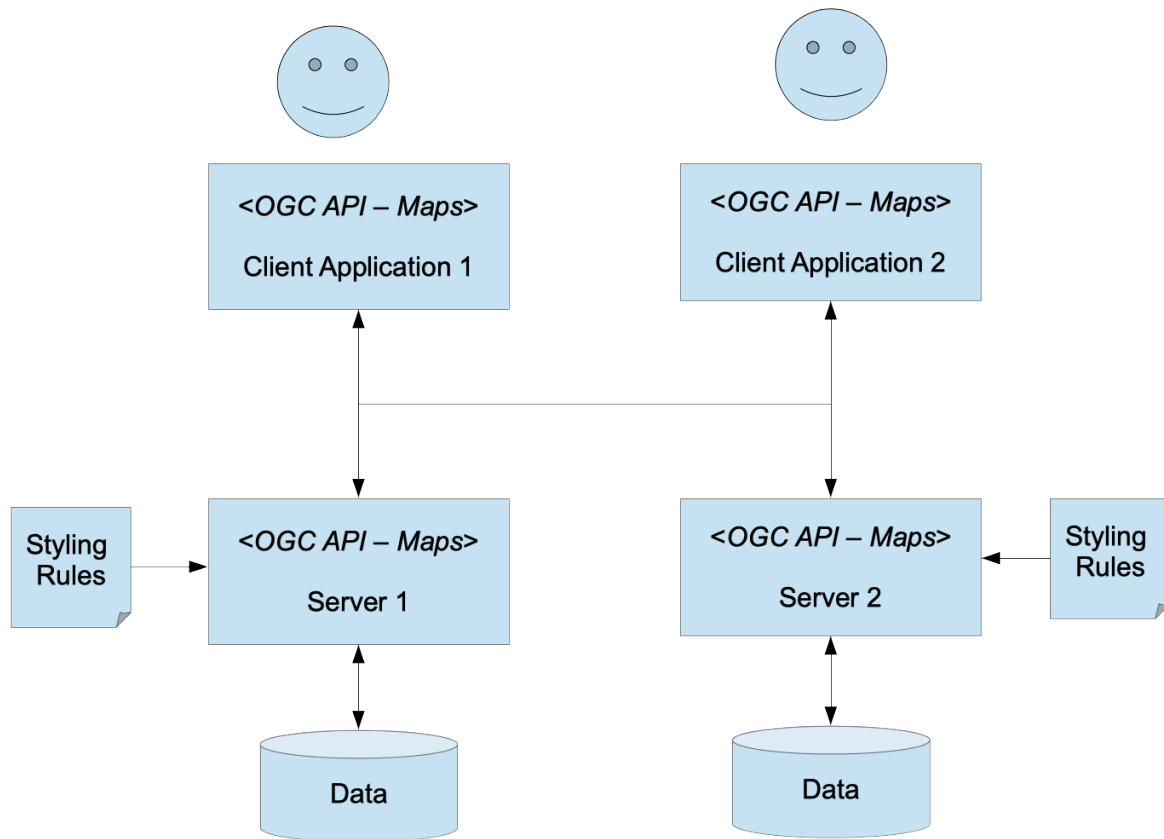


Figure 1. High level architecture implemented during the sprint

As illustrated above, the sprint architecture was designed with the view of enabling client applications to connect to different servers that implement OGC API - Maps. The servers were provisioned with data and styling rules, enabling them to render the data onto digital maps that were then sent to client applications in response to a GET request.

Chapter 7. Results

Multiple organizations provided servers, API implementations, and capabilities during the event. The remainder of this section describes each of the implementations.

The Centre for Ecological Research and Forestry Applications (CREAF) at the Autonomous University of Barcelona (UAB) deployed an instance of the MiraMon Map Server that implements support for the draft OGC API - Maps standard. The server is implemented as a CGI application encoded in C language as a part of the MiraMon Geographic Information System (GIS) & Remote Sensing (RS) suite and is interoperable with other vendors' clients. The user interface of the MiraMon suite is shown on [Figure 2](#).

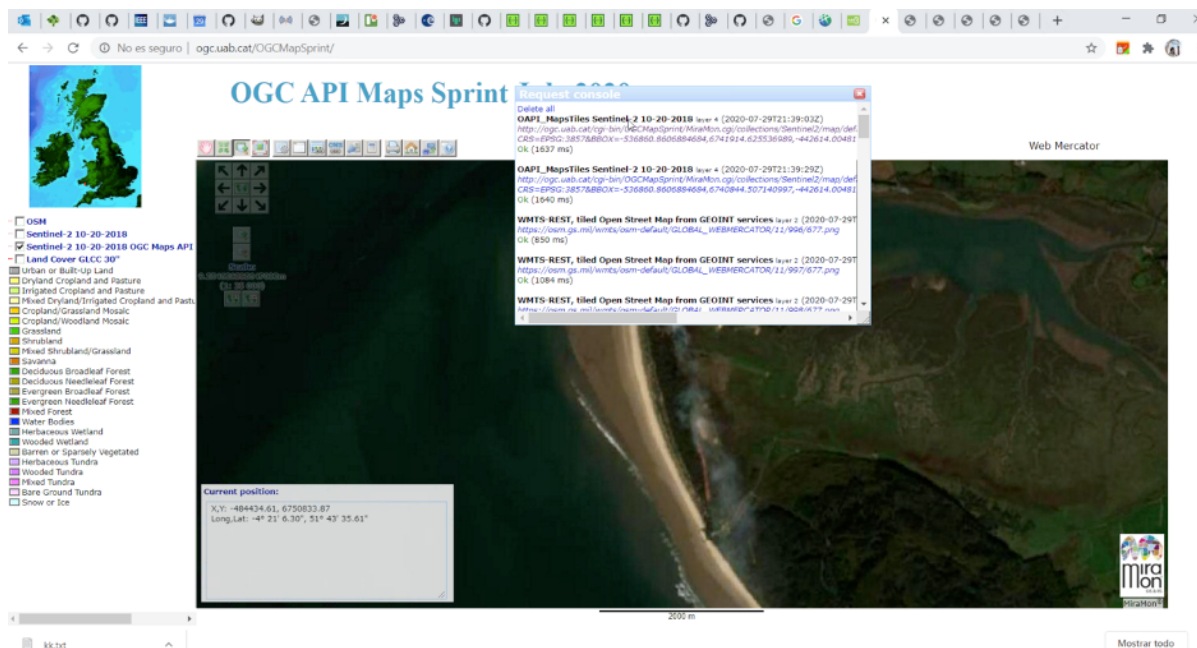


Figure 2. MiraMon client application by UAB-CREAF.

CubeWerx deployed an implementation of OGC API - Maps and other OGC APIs, CubeWerx server ("cubeserv"). The CubeWerx server was provided in the C programming language. The server exposes an interactive API description created from the OpenAPI definition document which enables both users and client applications to query the server. The server was updated in real time based on feedback from other participants using the services during the sprint. The API definition presented by the CubeWerx server is shown in [Figure 3](#).

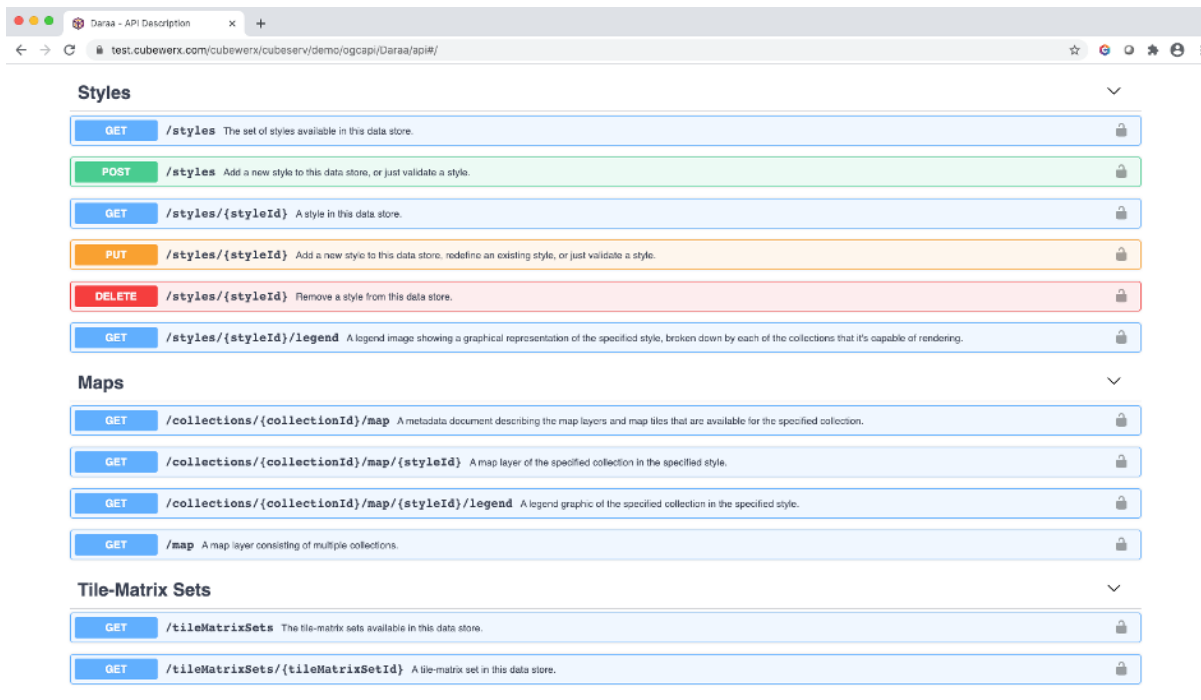


Figure 3. API definition presented by the CubeWerx server

Ecere initiated the development of server-side maps rendering capabilities, deployed at an endpoint supporting OGC API - Maps, and a number of other OGC APIs. The implementation was done in Ecere's GNOSIS Map Server product which leverages the open-source cross-platform Ecere SDK and eC programming language. An example output from Ecere's GNOSIS Map Server is shown in Figure 4.



Figure 4. A map produced by the Ecere implementation of OGC API - Maps

Ordnance Survey's OpenMapLocal product was among the maps published through Ecere's development server. A screenshot of a visualization of the data is shown in Figure 5. The OpenMapLocal product was exposed as a collection of maps through an OGC API - Maps interface. Ecere also worked on implementing client support for visualizing maps accessed through OGC API - Maps in GNOSIS Cartographer.

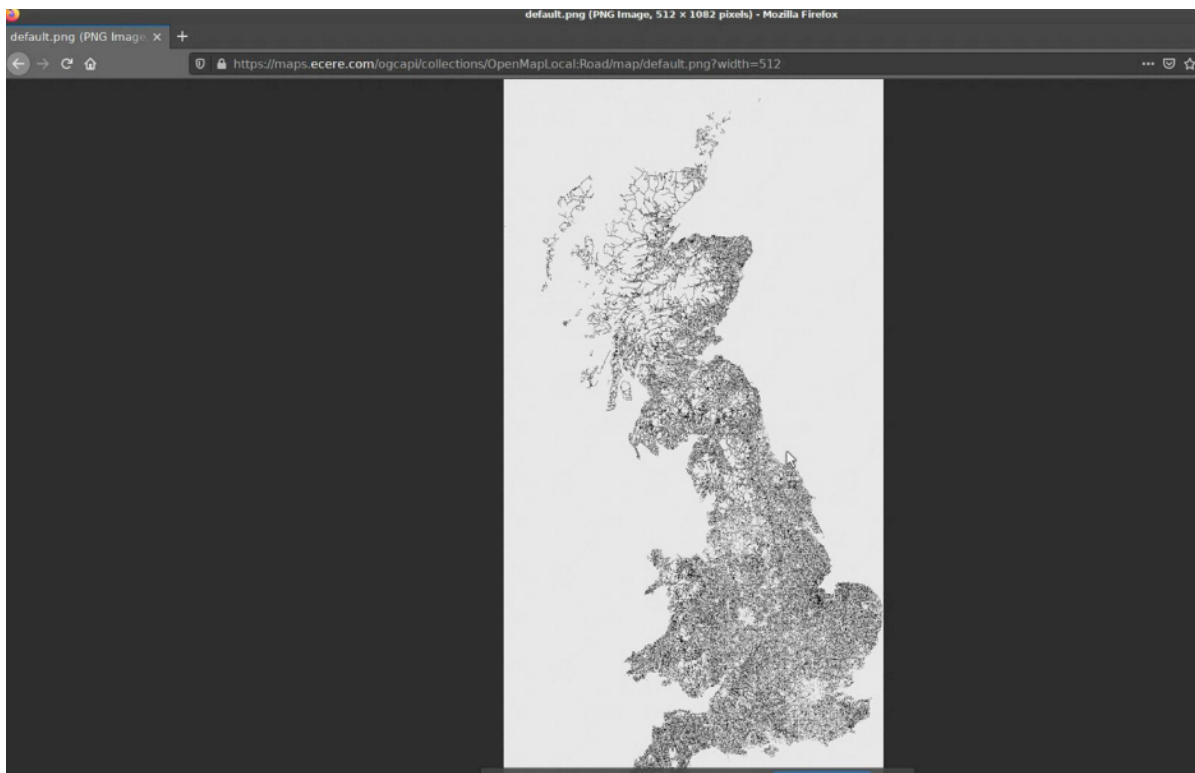


Figure 5. A map produced from OS OpenMapLocal data and rendered by the Ecere implementation of OGC API - Maps. Contains OS data © Crown Copyright and database right 2020.

Brad Hards, a participating developer, extended an existing OpenSphere plugin to support OGC API - Maps. OpenSphere is a pluggable, single-page geospatial web application that supports 2D and 3D views. OpenSphere supports binding to many popular servers and formats. Other features include animation of both raster and vector data, import and export of various formats, and saving files and layers between sessions. Figure 6 shows the OpenSphere application displaying a map retrieved from the Ecere implementation of OGC API - Maps.

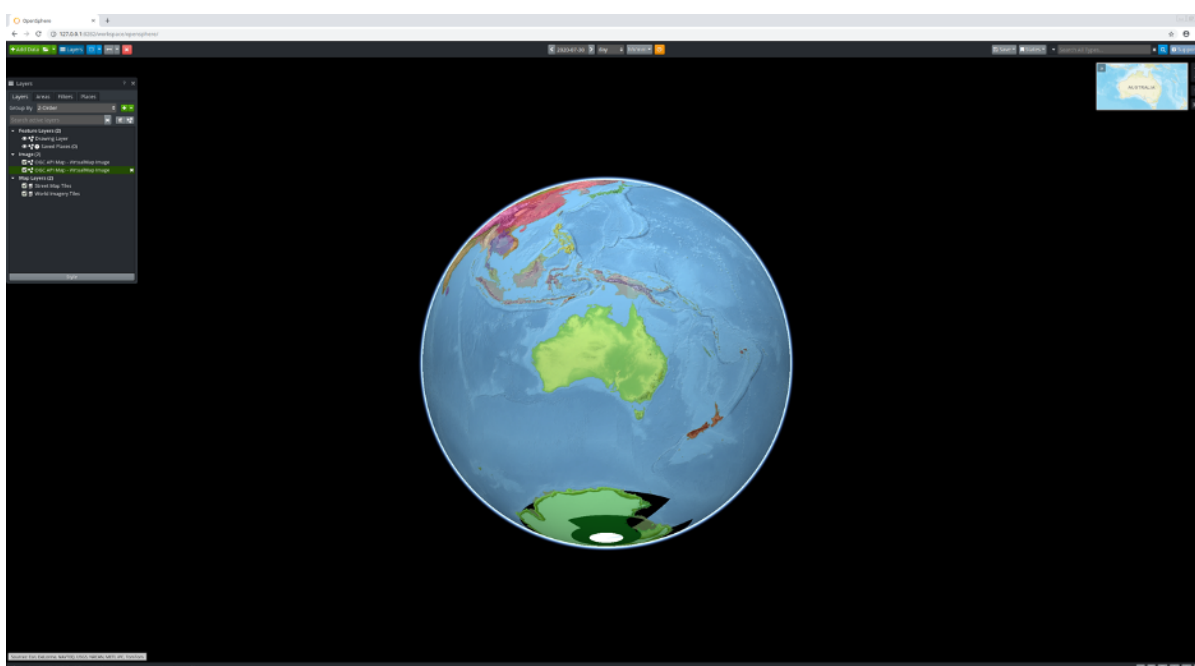


Figure 6. A visualization by the OpenSphere plugin extension created by Brad Hards

One of the prototype implementations, developed by OGC staff, demonstrated how to create an OGC API - Maps facade or proxy interface on top of a WMS. The WMS underneath the facade was

implemented using GeoServer. Development of the prototype facade was used as the basis for developing a [basic guide](https://github.com/opengeospatial/OGC-API-Sprint-July-2020/blob/master/docs/Draft_Spring_Guide_for_OGC_API_Maps/Draft_Spring_Guide_for_OGC_API_Maps.adoc) [https://github.com/opengeospatial/OGC-API-Sprint-July-2020/blob/master/docs/Draft_Spring_Guide_for_OGC_API_Maps/Draft_Spring_Guide_for_OGC_API_Maps.adoc] for developers that use the Sprint Framework (spring.io) - a popular framework for Java developers. The prototype implementation published a map, shown in [Figure 7](#), created from the Ordnance Survey's [OS Open Zoomstack](https://os.uk/business-government/products/open-zoomstack) [https://os.uk/business-government/products/open-zoomstack] product.

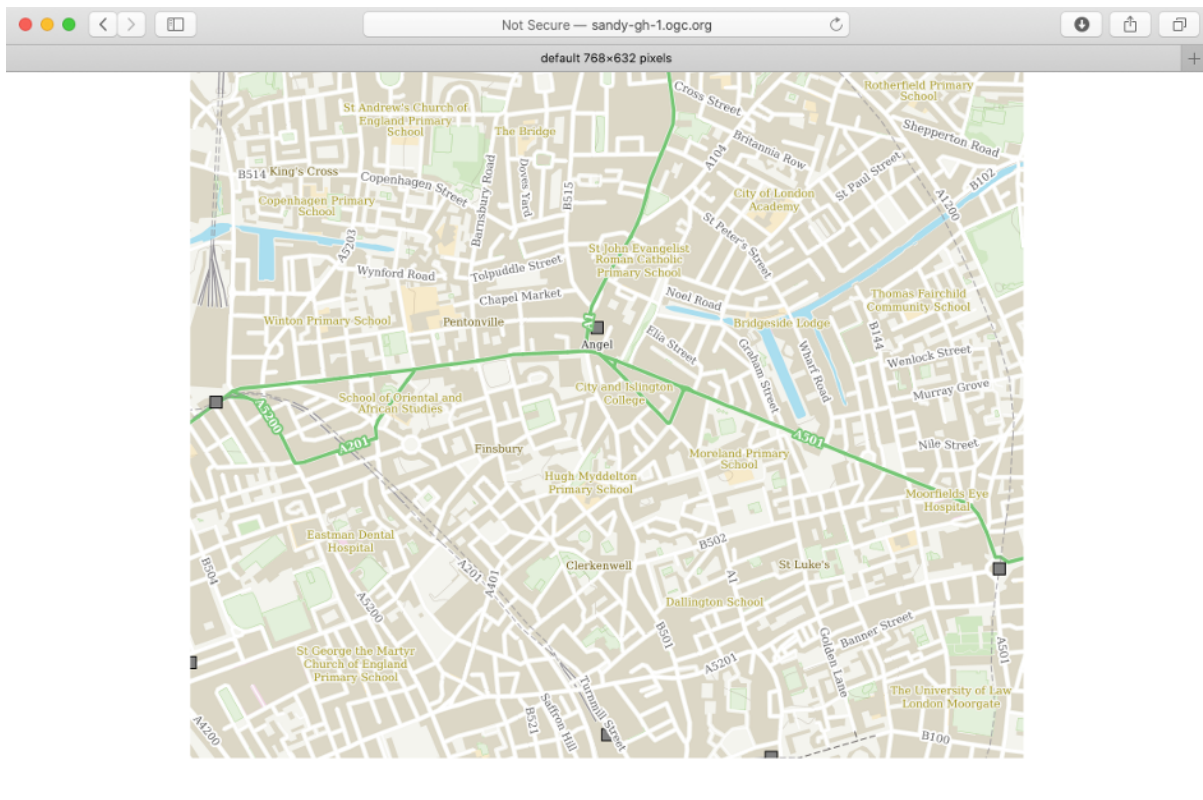


Figure 7. A map created from OS Zoomstack data and rendered by an OGC API - Maps facade on GeoServer. Contains OS data © Crown Copyright and database right 2020.

Chapter 8. Discussion

The sprint showed how JSON could be used successfully used to describe maps. In addition to JSON-based messaging, another significant difference between OGC API – Maps and a WMS instance was demonstrated during the sprint to be the ability to retrieve a complete map through a REST call without any query parameters. However, this capability raised the question of whether the standard should specify rules for default values of query parameters.

Early on in the sprint the participants agreed that an additional link relation type is needed to support the binding of client applications to server implementations of the OGC API – Maps draft standard. A number of link relation types were [proposed](https://github.com/opengeospatial/OGC-API-Maps/issues/48) during the sprint. The expectation is that upon approval of the standard, any new link relation types would be registered with the OGC Naming Authority (OGC-NA) in the [OGC Definitions Service](http://www.opengis.net/def/rel/) and ultimately with the Internet Assigned Numbers Authority (IANA).

Regarding styles and legends, the sprint participants [discussed](https://github.com/opengeospatial/OGC-API-Maps/issues/44) the need to specify a path for retrieving a legend. The role of styles and legends was also shown to be as important for OGC API – Maps as it has been for the WMS. This is an area where the emerging draft OGC API – Styles Standard will need to complement the functionality offered by OGC API – Maps [1].

The participants discussed a use case involving the use of custom styles sent to an implementation of OGC API – Maps. The participants explored how the following approaches, described in [Issue#42](https://github.com/opengeospatial/OGC-API-Maps/issues/42), could support such a use case:

- POST your style or execution document to `/map` with all query parameters and immediately retrieve the map image
- POST your style or execution document to `/map?mode=deferred` to get back a number of ways how you can query different AOIs/resolution (e.g. BBOX+width+height, tilesets...).
- POST your style or execution document to `/map/tiles/{tileMatrixSetId}` to get back an OGC API - Tiles tileset right away

There was acknowledgement across the sprint that the issue of posting styles was broader than just the Maps API, and that posting styles applies to the general posting of private resources. Therefore, the participants [agreed](https://github.com/opengeospatial/oapi_common/issues/159) that the issue should be addressed by the draft OGC API – Common Standard.

Another [issue](https://github.com/opengeospatial/OGC-API-Maps/issues/45) that was identified was whether the server should describe the spatial extents of each CRS, or whether that information should simply be inferred from the definitions of the CRS. If the spatial extent is inferred from the definition of the CRS referenced by identifier, then there would an expectation that the client application has access to the CRS definitions. Further, there was [discussion](https://github.com/opengeospatial/OGC-API-Maps/issues/49) about whether the CRS of the bounding box coordinates should be specified in a parameter called ‘crs’ or one called ‘bbox-crs’.

A copy of the draft standard was annotated with feedback from sprint participants. The participants expect that the editor of the draft standard will work with the SWG to revise the draft

standard according to the feedback provided. The rest of the discussion was captured on the Gitter platform. A screenshot of the Gitter channel is shown below. The Gitter channel can be found at <https://gitter.im/opengeospatial/OGC-API-Sprint-July-2020>

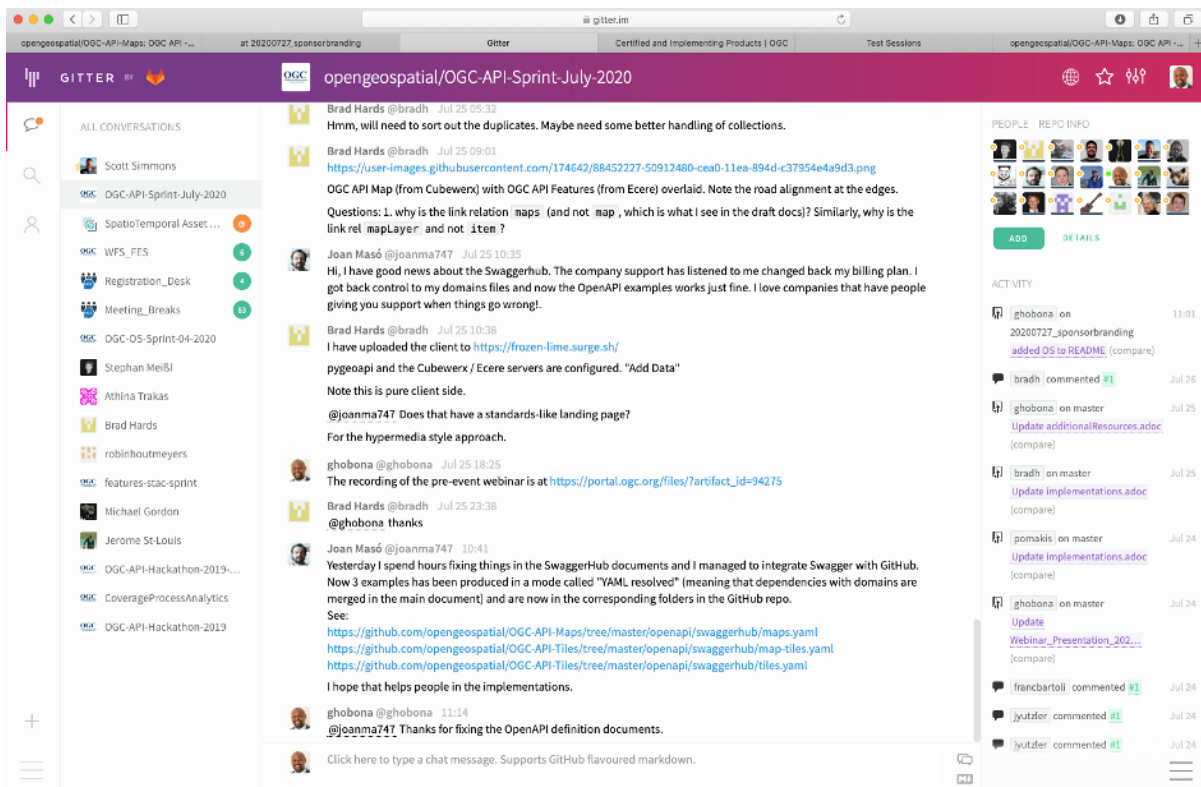


Figure 8. Screenshot of Gitter channel

A screenshot of the GitHub repository is shown below. The GitHub repository can be found at <https://github.com/opengeospatial/OGC-API-Sprint-July-2020>

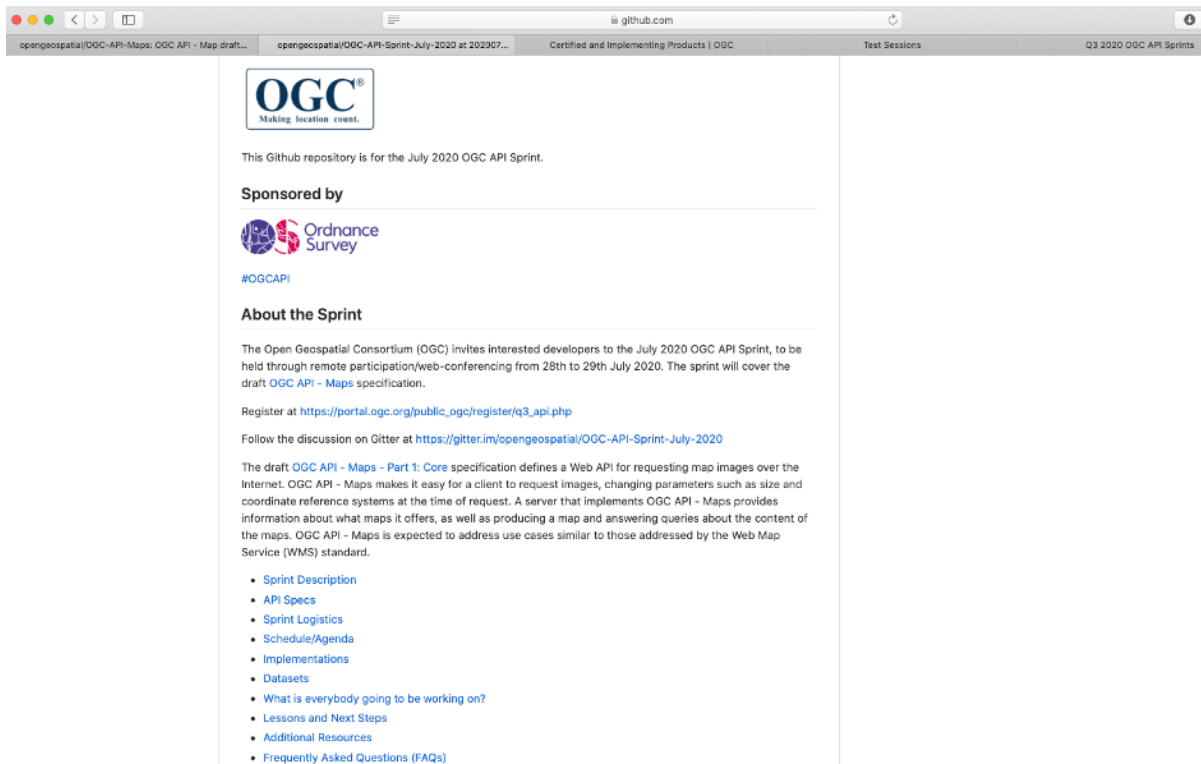


Figure 9. Screenshot of GitHub repository

8.1. Lessons Learned

Towards the end of the sprint, participants held a discussion on the lessons learned from the initiative. A summary of the lessons identified by the sprint participants is presented below:

- There is a need to provide a link to the legend of each map.
- There is a need for a 'types' property that implies that there is content negotiation at the *href* link. The format specified in 'type' is the default one. Currently if you have several formats, this results in a more complex encoding that has to repeat a full link for each format.
- There are several link relation types of map, legend, tileSet and so on. These could be useful for OGC API - Maps but are not available in the IANA register (<https://www.iana.org/assignments/link-relations/link-relations.xhtml>).
- Visualization of selected features could be achieved by returning geometry in a feature info response and then rendering that on the client side.
- There may be a need for a hierarchy whereby an application can organize collections according to a theme.
- OGC API - Maps is data centric. Therefore which dataset contributed to the creation of this map is clear. This is a significant benefit over WMS.
- The Maps API "knows" about items and coverages and is able to show maps. Showing processes is now easier.
- Using the OpenAPI definition, it is much easier to know what the server does without needing to read the specification. The OpenAPI definition clarifies information in the specification.
- There is still some institutional knowledge needed to understand the specification.
- A user guide could be useful to help developers get started.
- If there were a library that implements OGC API - Maps and everyone used that library, uptake would be accelerated.
- Making all the parameters optional is a significant improvement over previous approaches because this approach makes it easy to retrieve a map without providing any parameters.

Chapter 9. Conclusions

The sprint successfully demonstrated the ability of OGC API – Maps to publish digital maps, as well as the developer-friendliness of the draft standard. The sprint was the first to focus solely on the development of the draft OGC API – Maps Standard. Previous sprints and initiatives worked on both OGC API – Maps and OGC API – Tiles before the SWG’s decision to separate the requirements into two independent standards [2]. The successful implementation and demonstration of the draft standard during the sprint was therefore a significant achievement.

The participants demonstrated that the OGC API pattern can indeed address the needs of the map making, map publishing and map user communities. The participants envisage that as the draft standard continues to be developed, additional National Mapping Agencies (NMAs) will begin to familiarize themselves with the draft standard and the many benefits the draft standard could offer to them. There is, however, more work to be done in the SWG. Therefore the sprint reported in this document should be considered the first in a series of sprints that will be needed to move the draft standard closer towards approval by the OGC Membership.

9.1. Future work

The following general recommendations for future work items have been made.

9.1.1. Ideas for the Innovation Program

Future work in Innovation Program could include:

- A way to improve the querying. Query capabilities beyond feature info.
- Combining OGC API - Maps with processes and workflows.
- Continued progress of maps, styles, and tiles.
- Idea of geospatial data resources at different parts of the API. For example, if the SWG has to create a 'Collection of Collections' concept.
- To explore how addressing could be used to select bounding boxes (e.g. MGRS, What3Words, DGGS-based addressing) in a future extension. This may require a scale, a range, and so on.

9.1.2. Ideas for the Standards Program

Future work in the Standards Program could include:

- There’s a need to align the work of the OGC API - Maps SWG with that of the OGC API - Styles SWG.
- There’s a need to provide a guidance document that describes how to make use of elements from the OGC API - Common - Part 1: Core without necessarily pulling in all of the requirements of OGC API - Common.
- Work on Part 4: Custom Maps should be scoped.
- Consider the CRS of the Common and Features APIs, in relation to the BBox CRS.
- Geospatial data selection from a dataset could be sent to OGC API - Common.

Appendix A: Revision History

Table 1. Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
2020-11-12	G. Hobona	.1	all	initial version
2021-01-07	G. Hobona	.2	multiple	Revision based on feedback from Carl Reed

Appendix B: Bibliography

- [1] Portele, C.: OGC Testbed-15: Styles API Engineering Report. 19-010r2, Open Geospatial Consortium, <http://docs.opengeospatial.org/per/19-010r2.html> (2019).
- [2] Pau, J.M.: OGC Testbed-15: Maps and Tiles API Engineering Report. OGC 19-069, Open Geospatial Consortium, <http://docs.opengeospatial.org/per/19-069.html> (2020).