

Subject: Ongoing projects - status update as of 2024/02/21 / requests for clarification

From: Michal Siemaszko <mhs@into.software>

Date: 2/21/24, 15:52

To: Jürgen Albert <j.albert@data-in-motion.biz>, Mark Hoffmann <m.hoffmann@data-in-motion.biz>

Hi,

Status update as of 2024/02/21 is the following:

I. RE: "OGC API Features"

1. It was established that "Part 1: Core" (<https://docs.ogc.org/is/17-069r4/17-069r4.html>) and "Part 3: Filtering and the Common Query Language (CQL)" (<https://portal.ogc.org/files/96288>) of "OGC API Features" standard needs to be implemented.

What is not clear is which sub-sets of "CQL" should be implemented ? For a quick overview of "CQL2" sub-sets ("conformance classes" / "requirements classes") see: diagram @ <https://github.com/opengeospatial/ogcapi-features/blob/master/cql2/README.md>, also https://github.com/opengeospatial/ogcapi-features/blob/master/cql2/standard/clause_2_conformance.adoc

For example, "Basic CQL2" ? Which other sub-sets?

2. It was established that Lucene spatial search should be used on the backend. However, in "OGC API Features", what is used as "parameter" for such searches is a bounding box, whereas in one of the available implementations of Lucene spatial search (e.g. ``com.playertour.backend.golfcourse.CourseSearchServiceImpl.searchOrderedCoursesWithinDist(double, double, double)``), radius from a given latitude/longitude point is used.

This is not a problem, since instead of ``org.apache.lucene.document.LatLonPoint.newDistanceQuery(String, double, double, double)``, I can use ``org.apache.lucene.document.LatLonPoint.newBoxQuery(String, double, double, double, double)``.

However, what about indexing such shapes ? In ``com.playertour.backend.golfcourse.helper.CourseIndexHelper.mapCourse(GolfCourse, IndexActionType, Deferred<Boolean>)``, single point per golf course is indexed (golf course position) via ``org.apache.lucene.document.LatLonPoint.LatLonPoint`` and ``org.apache.lucene.document.LatLonDocValuesField.LatLonDocValuesField``. But GeoJSON, in addition to 'Point' geometry type, supports more complex geometry types such as 'LineString', 'Polygon', 'MultiPoint', 'MultiLineString' and 'MultiPolygon'.

Question is then: should only 'Point' geometry type be supported or more complex geometry types as well and if so, which and how should those be indexed via Lucene if only indexing of single points is supported .. e.g. each point separately via ``org.apache.lucene.document.LatLonPoint.LatLonPoint`` and ``org.apache.lucene.document.LatLonDocValuesField.LatLonDocValuesField`` as part of a given document ?

II. RE: "Avatar"

On February 6th, I provided a fully working solution via PR <https://github.com/DataInMotion/avatar-libraries/pull/2/> along with technical documentation outlining problems and solutions applied to those problems, in addition to list of questions.

It appears that Mark "jumped" straight to questions, without checking technical documentation provided, and perhaps not checking working solution provided via this PR.

For each question asked, Mark created an issue (<https://github.com/DataInMotion/avatar-libraries/issues?q=>), where I had to repeat multiple times information provided via technical documentation section of that PR.

I outlined in detail the different options checked and problems which arose, including ``loadInitialization="true" / `loadInitialization="false"` and many, many other.

I captured logs, I took screenshots - I can share this as supplementary information, if technical documentation is not enough (provided it is actually read).

Since I shared a working solution via that PR, as part of that I had to introduce things such as ID fields, etc.. All this was done so working solution can be provided..

Instead of providing comprehensive answers to questions I asked, so I can apply these changes to PR I shared and libraries used underneath, I had to repeat things over and over again, and Mark committed directly to develop branch his version, after, as he mentioned "updating some dependencies", and introducing things which I already introduced in PR shared (e.g. ``extId`` field, etc.).

This is no way to solve problems and supplement knowledge. Especially if there's an issue in underlying library, which is fixed and "updated", I need to know where the problems at a given point in time stemmed from, not having them silently corrected without explanation..

I would like to clarify what should be the next steps on this project then and in what format should I provide technical documentation and working solutions, so those are familiarized with before answering questions.

Regards,

--

Michael H. Siemaszko
Mobile: +48 668 566 023
Telegram: mhsiemaszko
Email: mhs@into.software,
mhsiemaszko@7thraylabs.com
WWW: <http://ideas.into.software/>
GitHub: <https://github.com/ideas-into-software/>
LinkedIn: <http://www.linkedin.com/in/mhsiemaszko/>
Twitter: <https://twitter.com/IntoSoftware/>