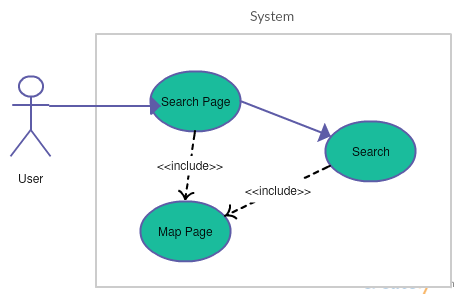
**YTU-CE: Semantic Web Graduate Course   
Fall 2016   
Term Project  
11501203 Ali Alperen ÇOLAK**

This term project aims to create a substructure and a view point for generating geospatial ontology of Turkey beginning with Bozcaada- an Aegean touristic island where is famous for its beaches, historical places and grapes- the only district in the country that has not any villages.

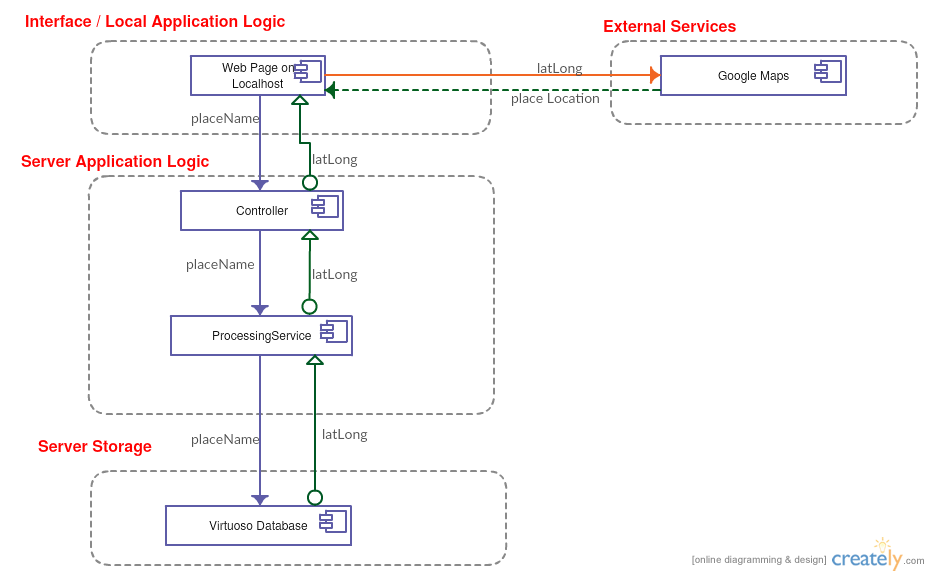
**Use Case Scenario**

User searches a place from search page which is located at <http://localhost:8080>. System searches the location and returns location of the place on the map page. User can continue the search process in this page.

****

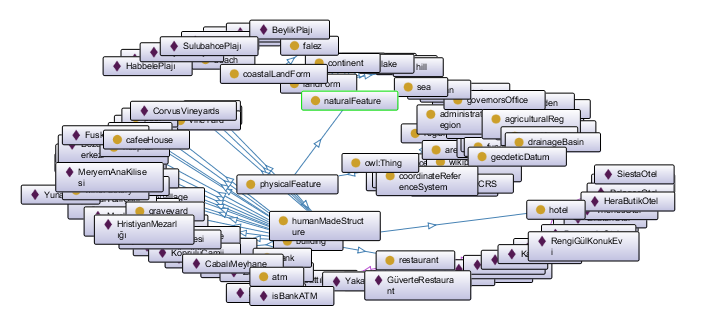
**System Architecture**

The system stands on two servers. Tomcat server for hosting web files and Virtuoso server for hosting data. Tomcat server is online at port 8080 and Virtuoso http server is online at port 8890. Search string placeName input sent by web page to Spring controller. Controller catches the request and sends it to service. Service creates a query with Jena provider and sends it to Virtuoso server. Virtuoso server returns results and service gets latitude and longitude values of the place. These values send backwards to controller. Then controller sends them to web page. Web page requests a map from Google Map Api with these values (latitude and longitude) and shows it.

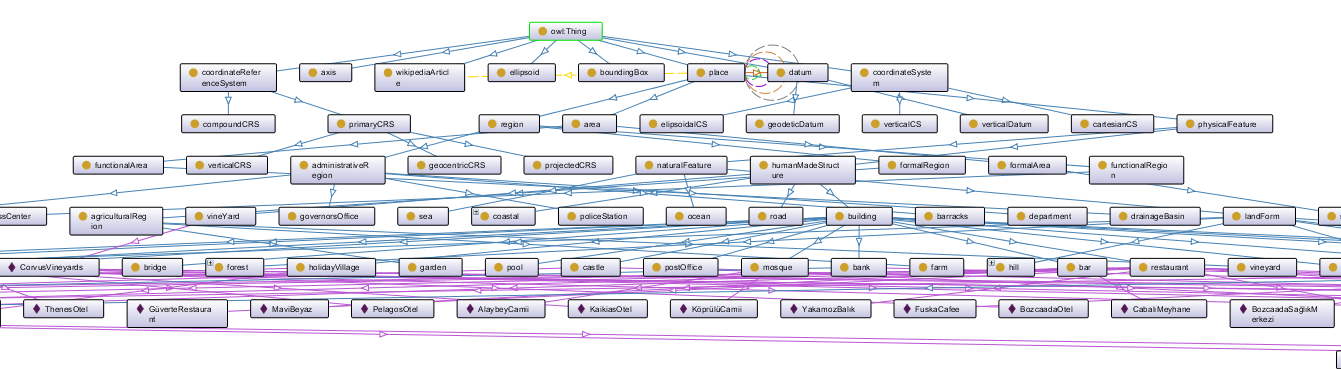
****

**Class Hierarchies**

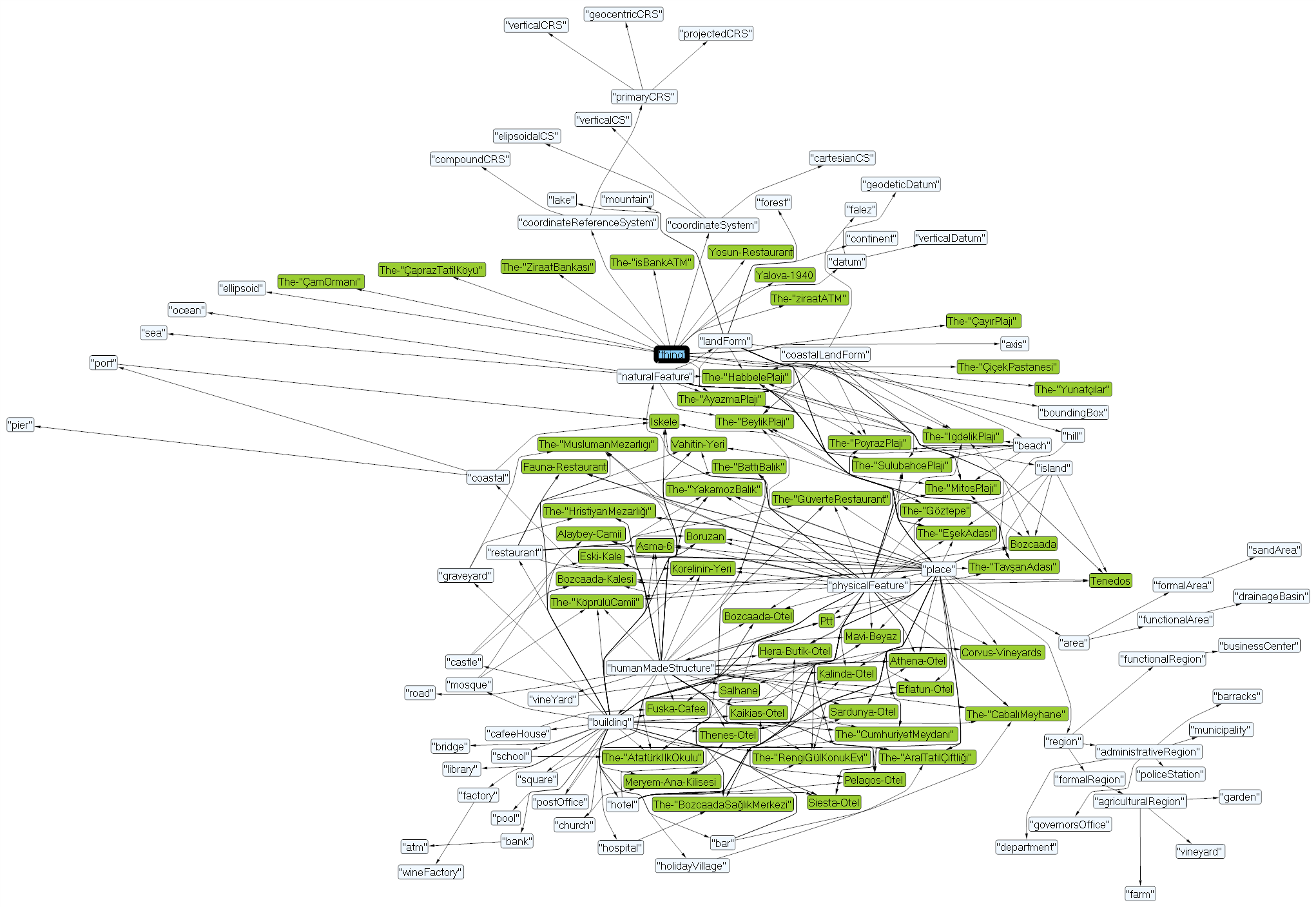
Centered form:

****

Tree form:



Document diagram created with Ontorion

****

**Technologies, Technical Challenges and Brief Explanation**

Geological data has very different types. Geological representation of a location varies due to used coordinate reference system. Geocentric, geographic and projected coordinate systems are used in the area. Knowing Google Maps Api uses the geographic (vertical) coordinate system is essential. In the vertical system latitude and longitude values represents parallel and meridian values of the place. Thus places are shown with a marker on the map. Although ontological classes designed to provide “touches”, ”wiki Address”, “is part of” and “overlaps” relations and places can be shown with polygons, implementation of them is left to further studies. Ontology created in Protégé. The ontology contains 508 axioms, 354 Logical axioms, 78 classes, 7 object properties, 10 data properties, 59 individuals. Ontology contains classes that define different kinds of places; places that are either formed by nature (island, beach, etc.) or defined by humans (hotel, restaurant, etc.). The instance data is derived from Google Maps.

Virtuoso is used for ontology data server. Linked data is added to server as conductor role as Quad Store with <http://localhost:8890/geolocation> Named Graph IRI. Jena is used for Java provider of Virtuoso. Sparql is used for querying the ontology. But Jena and Virtuoso have different dialects for it. To generalize Sparql query <http://sparql.org/validate/query> generator is used. Besides that, to use Jena query with Virtuoso server, query must be sent to <http://localhost:8890/sparql> address.

To reach Google Maps Api and represent desired location on Google Map, the project created as an enterprise application. This is achieved with using Spring Boot for project architecture. Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run". With Spring boot user does not need to establish a Tomcat Server because the project consists it. User interfaces as web pages hosted by Tomcat Server which is available with the project run on <http://localhost:8080/> . To run the project appropriately, port 8080 must be free before the run.

Thymeleaf is used for creating user interfaces. Thymeleaf is a modern server-side Java template engine for both web and standalone environments. Thymeleaf has html-like syntax and creates web pages according to html files on the fly. One problem with Google Maps Api integration is that developer has to find a way to add JavaScript functions, external JavaScript library, a Google Map and a marker for the location in a thymeleaf file. The problem can be easily overcome by following instructions on documentation.

To use Google Maps Api in the application, it is required to have a Google developer account and create a Map key to use map. Application sends requests to Google with this key. In the application Google Map JavaScript Api is used.

**Experience and Discussion**

Geological representation is a very deep area and its implementation needs reaching land registry information. Polygonal representation and mereological and topological relations requires further research and data. In the project user can search places in an ontology via an interface and the system finds latitude and longitude values of the location and shows the location of the place on a Google Map.

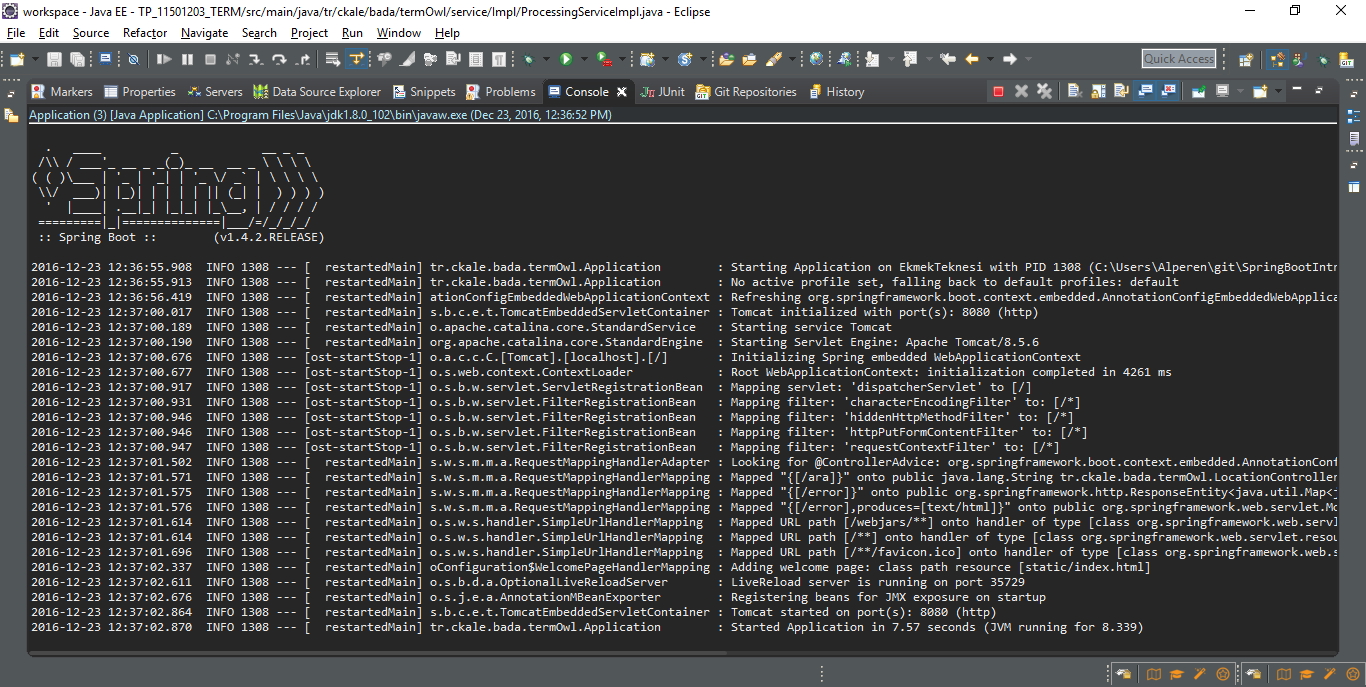
**How to Compile and Run**

First, Virtuoso must be installed on the device. Virtuoso server must be run with virtuoso\bin>Virtuoso-t -f command. Then geolocation.owl (located in owlData) must be imported to Virtuoso by Linked Data – Quad Store Upload menu (<http://localhost:8890/conductor/rdf_import.vspx?sid=d4b05bbde43a3b5a6ea0669c2e2d10ee&realm=virtuoso_admin>) with <http://localhost:8890/geolocation> as Named Graph IRI.

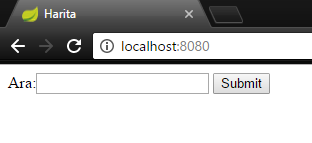
The project is a JEE project. Although dependencies managed by Maven, Virtuoso jars must be manually added to project build path, since there is not any repository for Virtuoso.

To run the project right click to Application.java (located in tr.ckale.bada.termOwl package) Run As- Java Application.

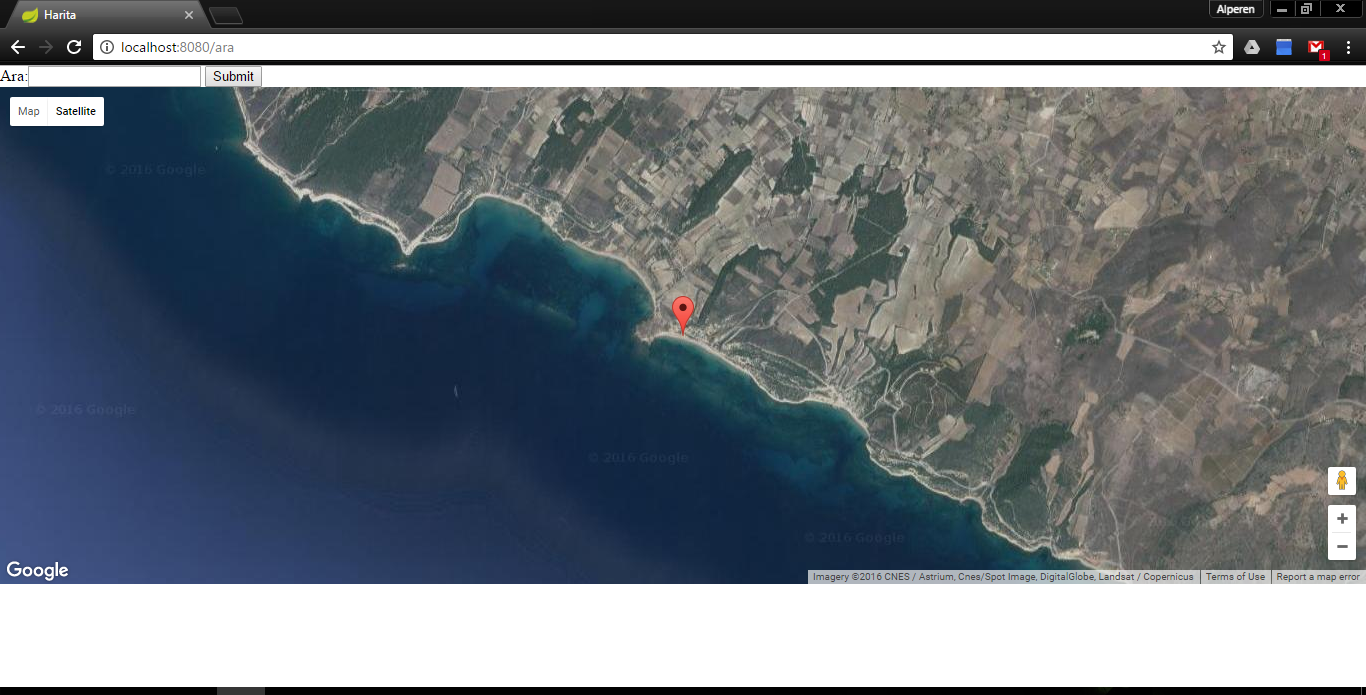
This is console view after running the project.



After that, the system can be reached at <http://localhost:8080/>



Suppose that “Ayazma” is searced, the response screen is like follows:



**REFERENCES**

<http://seco.cs.aalto.fi/applications/geosearch/>  
<http://demo.seco.tkk.fi/onkipaikka/>  
<http://seco.cs.aalto.fi/publications/2008/henriksson-kauppinen-hyvonen-suo-2008.pdf>  
<https://ahmedchan.wordpress.com/2014/02/19/virtuoso-servers-installation-setup-data-upload-and-querying-on-windows/>  
<http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtRDFInsert>.  
<http://sparql.org/validate/query>   
<http://nixmash.com/java/using-google-maps-api-with-thymeleaf-and-spring/> <http://stackoverflow.com/questions/40746528/google-map-not-showing-spring-maven-thymeleaf-salespoint>  
<https://projects.spring.io/spring-boot/>