

0001 AtmAssignment

Document the understanding of the assignment (for a SA position at Backbase) sent by Janet Casteelen
Source code is let available at: https://github.com/andhdo/0001_atmassignment.git . It also contains a copy of [this document](#) .

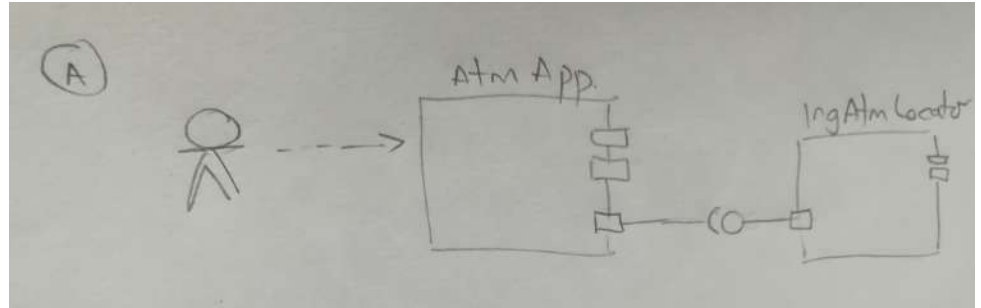
By AndresHurtado (andhdo@gmail.com)

WORK IN PROGRESS

Architecture Extract

Conceptual View

We need to show ING ATM locations to the end user through a web application because his locality. According the instructions document as a source of requirements, we need to filter a document exposed by a rest remote service and filter those messages by a city name. Then we need to consume this filtered service and show this payload to the final user.



Development View

Guided by the NFRs¹ included inside the requirements we can build a more detailed static view of what we're trying to build:

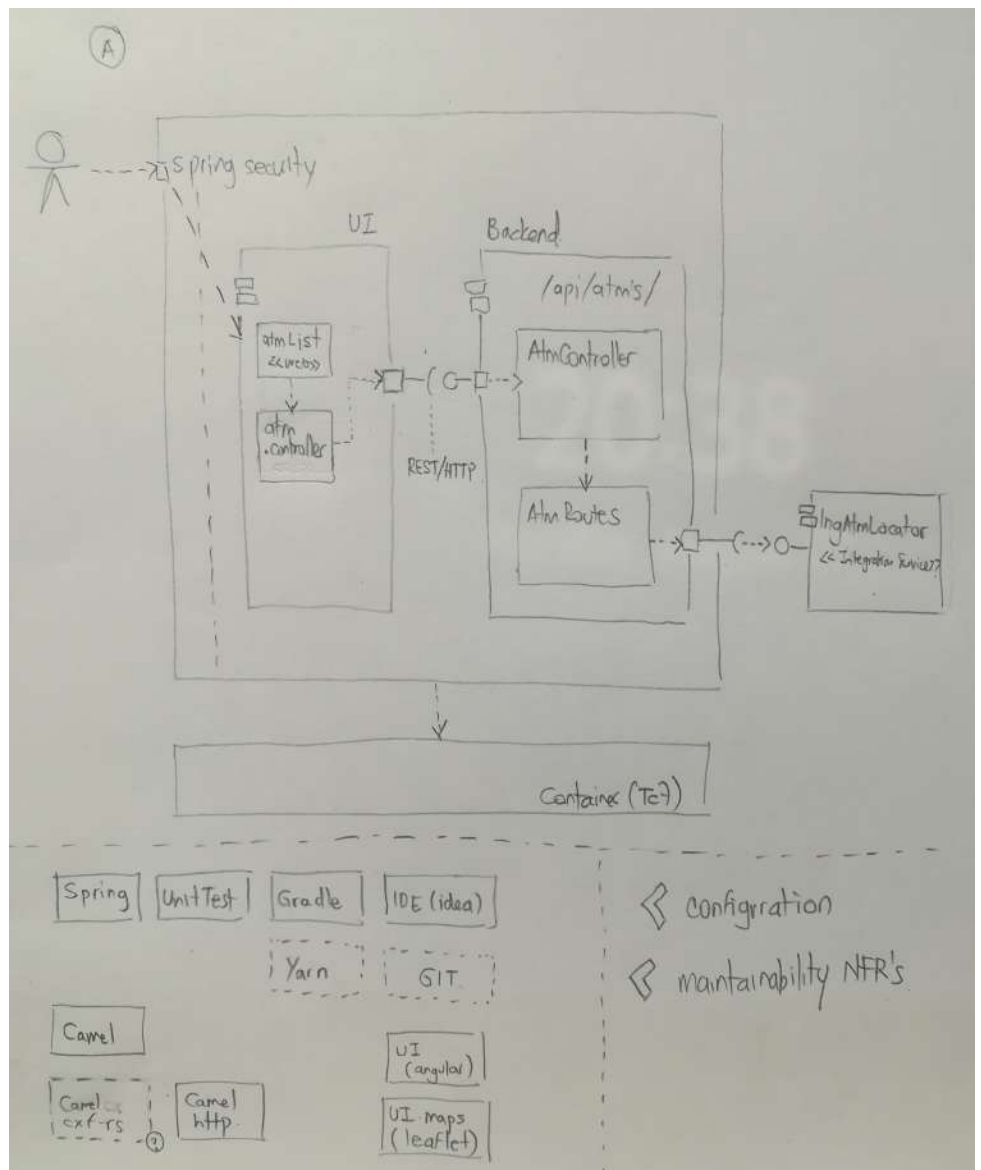
We need to build a spring application within a version control system (git), it's a web application that needs a build system (gradle), and an IDE to develop (idea-intellij). Then we need a way to consume and filter the rest service exposed by ING (the ATM list service), and some way to visualize the results.

With those requirements in hand, mainly I have two alternatives for the construction:

- Spring initializr
- jhipster

The second one is more heavyweight, including other tools like migrations and ORM framework and microservices capabilities that are not needed for this project. Then we are going to go with the first one as a simple monolithic application.

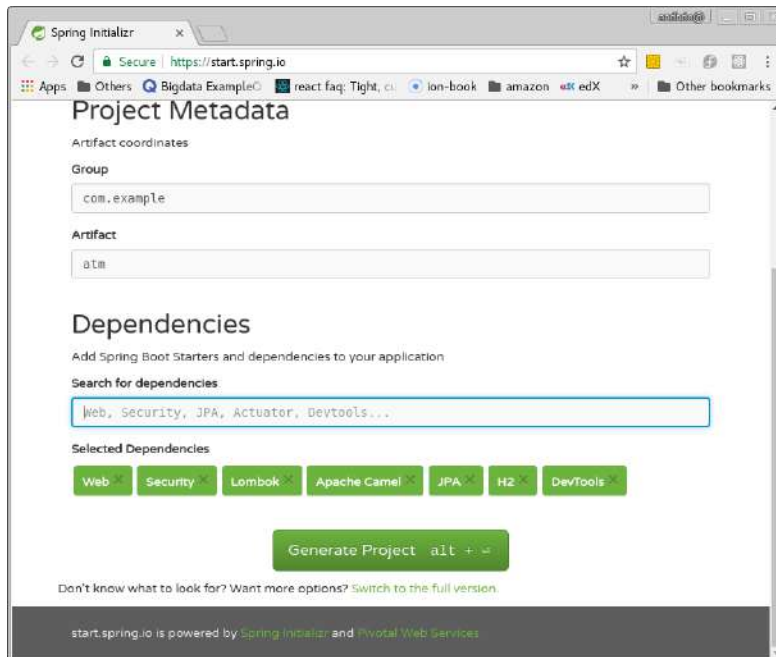
The application stack to be used is based in spring web, because it can provides us integration with spring-security and then we can access the payload processed by camel. In



¹ NonFunctional Requirements

this scope, camel can be seen as a lightweight integration framework that we can embed in the application and manipulate to do the integration with the ING atm service. In the case of java, initializr can give us a simple skeleton of the application. Spring-boot is used as basis to leverage the construction of the application instead of <xml> spring old legacy configuration.

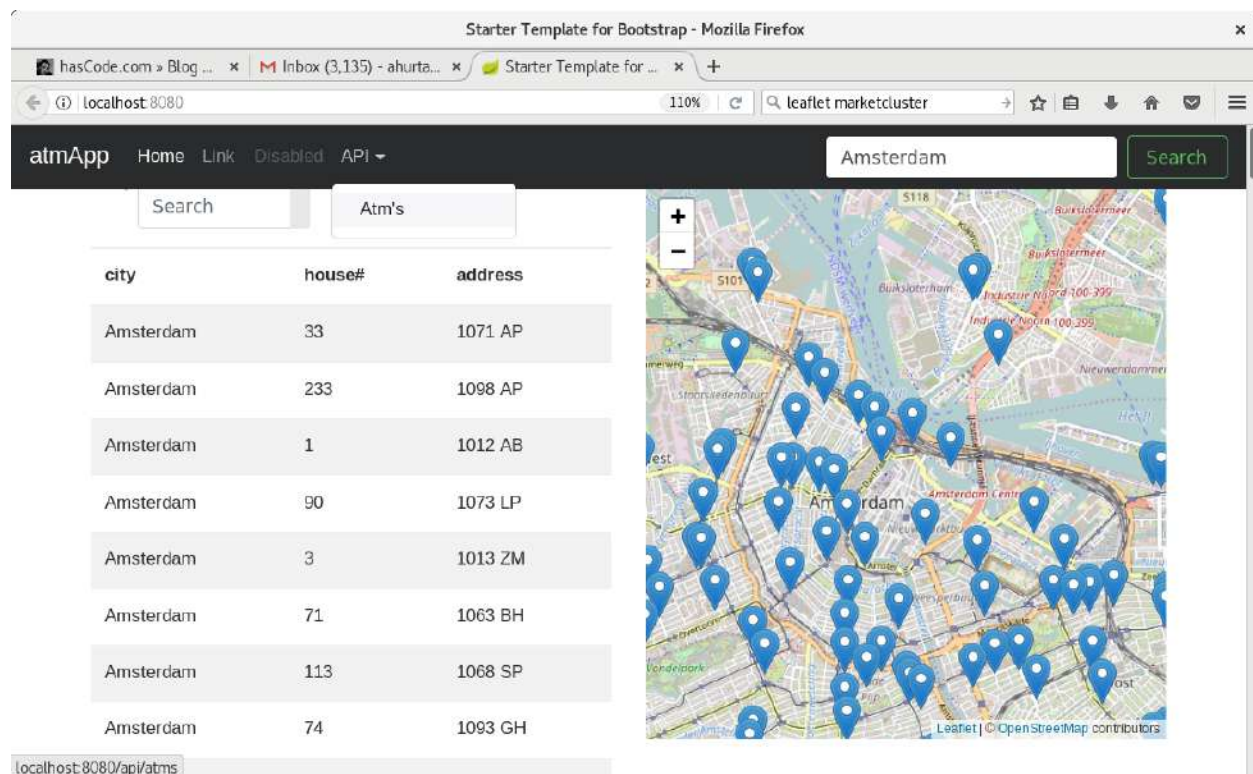
We also need to install all the plumbing to build the web application that includes IDE, git-client tool, gradle (build tool)....



Because i'm not working in projects of java at this moment, we can look for a simple tutorial of spring boot and camel to build the baseline of the project. Also is not a bad idea to look for a tutorial to remember angular 1.0 that some time ago i used before and it fits with spring-boot out of the box (or at least is the promise of jhipster framework). Also this can help to lower the tools needed inside the ui (yarn,ts-preprocessors,...) in comparison of angular-4 (but maybe a future release of the application can include it).

For the UI we need to consume the service and then display the result of the service. We can include some angular and bootstrap libraries and make a simple layout for the project, and then use it to consume and preprocess the rest service for visualization purposes.

The final result can be seen in the image below, where I have included leaflet library and directive to show the payload.



Through a rest controller I'm exposing the rest results, but at this time i'm not passing a filter parameter to the request

```

Mozila Firefox
hasCode.com » Blog ... x  Inbox (3,135) - ahurta... x  http://loca...80/api/atms x +
localhost:8080/api/atms 110% leaflet marketcluster
[{"address":{"street":"Van Baerlestraat","houseNumber":"33","postalCode":"1071 AP","city":"Amsterdam","geoLocation":{"lat":"52.357087","lng":"4.879845"},"distance":0,"type":"ING"},"address":{"street":"Middenweg","houseNumber":"233","postalCode":"1098 AP","city":"Amsterdam","geoLocation":{"lat":"52.349501","lng":"4.937735"},"distance":0,"type":"ING"},"address":{"street":"Stationsplein","houseNumber":"1","postalCode":"1012 AB","city":"Amsterdam","geoLocation":{"lat":"52.377907","lng":"4.901844"},"distance":0,"type":"ING"},"address":{"street":"Van Woustraat","houseNumber":"90","postalCode":"1073 LP","city":"Amsterdam","geoLocation":{"lat":"52.354964","lng":"4.900921"},"distance":0,"type":"ING"},"address":{"street":"Spaarndammerdijk","houseNumber":"3","postalCode":"1013 ZM","city":"Amsterdam","geoLocation":{"lat":"52.39165","lng":"4.877995"},"distance":0,"type":"ING"},"address":{"street":"Burgemeester de Vlugtlaan","houseNumber":"71","postalCode":"1063 BH","city":"Amsterdam","geoLocation":{"lat":"52.379665","lng":"4.83135"},"distance":0,"type":"ING"},"address":{"street":"Osdorpplein","houseNumber":"113","postalCode":"1068 SP","city":"Amsterdam","geoLocation":{"lat":"52.359387","lng":"4.806997"},"distance":0,"type":"ING"},"address":{"street":"Eerste van Swindenstraat","houseNumber":"74","postalCode":"1093 GH","city":"Amsterdam","geoLocation":{"lat":"52.362632","lng":"4.928015"},"distance":0,"type":"ING"},"address":{"street":"Arent Janszoon Ernststraat","houseNumber":"551-559","postalCode":"1082 LD","city":"Amsterdam","geoLocation":{"lat":"52.3321007","lng":"4.8775673"},"distance":0,"type":"ING"},"address":{"street":"Oostelijke Handelskade","houseNumber":"1065","postalCode":"1019 BW","city":"Amsterdam","geoLocation":{"lat":"52.373962","lng":"4.938648"},"distance":0,"type":"ING"},"address":{"street":"Van Woustraat","houseNumber":"150","postalCode":"1073 LV","city":"Amsterdam","geoLocation":{"lat":"52.353796","lng":"4.903047"},"distance":0,"type":"ING"},"address":{"street":"Hugo de Grootplein","houseNumber":"17","postalCode":"1052 KV","city":"Amsterdam","geoLocation":{"lat":"52.374657","lng":"4.872771"},"distance":0,"type":"ING"},"address":{"street":"Beukenplein","houseNumber":"6","postalCode":"1091 KG","city":"Amsterdam","geoLocation":{"lat":"52.357386","lng":"4.917528"},"distance":0,"type":"ING"},"address":{"street":"Jodenbreestraat","houseNumber":"21","postalCode":"1011 NH","city":"Amsterdam","geoLocation":{"lat":"52.368827","lng":"4.903603"},"distance":0,"type":"ING"},"address":{"street":"Buitenveldertselaan","houseNumber":"164","postalCode":"1081 AC","city":"Amsterdam","geoLocation":{"lat":"52.325261","lng":"4.867929"},"distance":0,"type":"ING"},"address":{"street":"Van der

```

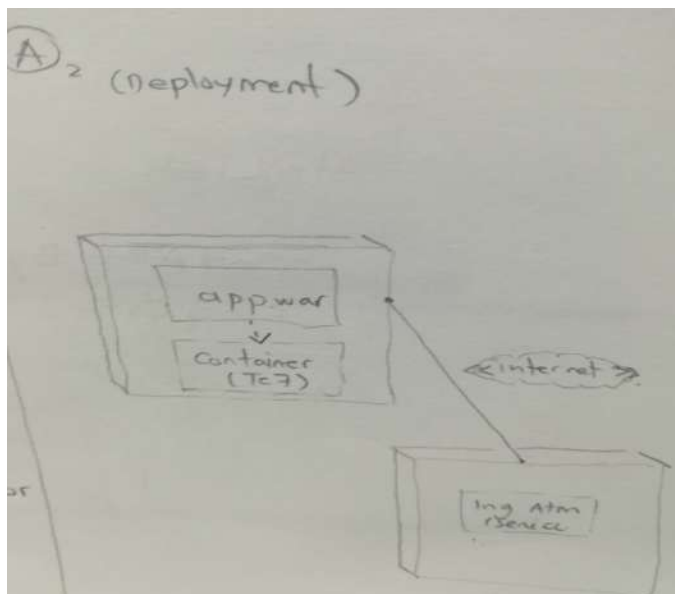
However inside the camel route, I'm doing the filter for the Amsterdam city (currently burned in the code):

```

AtmController.java  AtmRoutes.java  BasicController.java  sample_response.jsonp
import ...
16
17 @Component
18 public class AtmRoutes extends RouteBuilder {
19
20     @Override
21     public void configure() throws Exception {
22
23         // to verify camel integration
24
25         from("direct:home")
26             .log("Camel body: ${body}")
27             ;
28
29         from("direct:atmList")
30             // .log("original ${body}")
31             .transform()
32             .jsonpath(text: "${?(@.address.city=='Amsterdam')}")
33             // .log("filteredData [result: ${body}]")
34             ;
35
36         // to verify routes
37         from("direct:atmListFile")
38             .process(new Processor() {
39
40             @Override
41             public void process(Exchange exchange) throws Exception {
42                 Path sampleFile = Paths.get("src/main/resources/sample_response.jsonp");
43                 File file = sampleFile.toFile();
44
45             }
46
47         });
48
49     }
50
51     AtmRoutes() { configure(); }
52
53 }

```

At this time I'm consuming a file with the results, because a problem of protection that the ING service has (see the [Comments](#) section).



Deployment View

The application deployment as is, can be packaged as a jar file with an embedded tomcat container, or can be packaged as a war. Inside the README.md file of the project there are instructions to build it.


Resulting war file is almost 27 Mb, because some camel additional libraries to parse json

Project Management

Above we have a description of the activities and an approximate of the time consumption

	Activity	Time	Estimated
	Download and installation of tools	2.5h	
	Architecture and Design (DLD)	1.5h	
	Project baseline (Coding)	2.5h	
	Tutorials and remembering spring-boot, angular, apache-camel	2h	
	Error of json protection of original file	1.5h	
	UI of the project (Coding)	2h	
	Document creation and description	2h	
	Total time	12.5h	5h

Comments

- For protection, the ing service manages a common prefix ("  "). I created a filter for this because in another way the json couldn't be manipulated easily by camel, but couldn't include it because of the time. It was a waste of time because tools like postman (<https://www.getpostman.com/postman>) hides the detail of the invoked payload.

References

Change Control

Date	Version	Comments
2017.08.07	1.0	Initial version [AH]
2017.08.07	1.1	Add project mgmt details