Apache Tomcat as a transport in CXF

Initial plan

Team and contact information:

Ivan Kravchenko (ivan.kravchenko@students.unibe.ch)

Guillaume Corsini (guillaume.corsini@students.unibe.ch)

23. March 2019

Version 1.3

Table of contents

Version monitoring

Monitoring of the version in this project organization document enables following modification activities progress within the project. The version is changed only if a major change takes place. Otherwise, only a decimal is changed (1.0 to 1.1)

Version	Update date	Page modified
1.0	14. March 2019	new document
1.1	17. March 2019	5, 6, 9 & 10
1.2	21. March 2019	4, 7, 8 & 9
1.3	23. March 2019	4, 6, 8 & 9

1. Project description

CXF is an open-source framework, which simplifies building web services. Tomcat is a popular web server running a Java Virtual Machine (JVM). The project intends to enable CXF to use an embedded Tomcat server for transporting HTTP requests. In the following chapters, the use of the words “embedded Tomcat” will be shortened by writing only “Tomcat”.

1.1. Project context

Red Hat is a company providing many services for enterprises mainly based on open source software. As promoters of open source code, they publish all of their developed code. One of the many projects that Red Hat is actively developing is the Tomcat server. The tomcat server is an open source web server for Java programming language. In order for it to continue being popular in the future, Red Hat wants to have it integrated into the Apache CXF framework. Our project should enable the connection of these two tools.

1.2. Goals and objectives of the project

The major goal is to deliver a proof of concept that Tomcat can be used as a transport in CXF similarly to Jetty or Undertow. This will be achieved by reaching the following objectives:

1. Study and understand the following technologies: CXF (especially transport requirements), how embedded Jetty and Undertow servers are used as transports in CXF.
2. Study how an embedded Tomcat can be added to the CXF framework.
3. Propose a concept resulting in an embedded Tomcat being used as a transport in CXF.
4. Implement the aforementioned concept using a test driven approach.
5. Provide a proof of concept by developing a simple application which uses an embedded Tomcat as a transport in CXF.

A secondary goal is to complete the implementation of the aforementioned concept in order to deliver a fully functional transport in CXF. This goal is not a requirement, but will be tackled if there is still time left for the students.

Finally, another important goal is to write detailed documentation ensuring that others could take over the project in order to finish all the implementations or to make further adaptations.

2. Project organization

We are a group of two students that are supported by a project supervisor from Red Hat and a workshop coordinator from the university of Neuchatel. Each student will work two days a week on this project. Most meetings will be held online using BlueJeans. Coordination of tasks between team members is done using a Trello board.

2.1. Responsibility distribution

- Ivan Kravchenko; student at university of Berne
  ivan.kravchenko@students.unibe.ch

- Guillaume Corsini; student at university of Berne
  guillaume.corsini@students.unibe.ch

- Jean-Frédéric Clere; project supervisor at Red Hat
  jclere@redhat.com

- Dr. Hugues Mercier; workshop coordinator at university of Neuchatel
  hugues.mercier@gmail.com

2.2. Interactions with the client

The team members interact by email and mobile messaging. Additionally, a project board on Trello will be used to simplify the coordination between the team members. The complete codebase will be hosted publicly on github. Most meetings with the client will be held online using BlueJeans. For questions or issues, the client will be contacted by email. Red Hat will ensure that the minimal standard of performance required by Tomcat and CXF are met. Finally, weekly status updates will be sent to the client by email.

3. Methodology

This project will follow the framework proposed by “Design Science Research in Information Systems” from Von Alan, R. Hevner, et al. in *MIS quarterly* 28.1 (2004): 75-105. Hence, we will study the existing systems, propose a concept, develop the solution and finally test it. The concept will also provide the minimal requirements to produce a working demonstration. Developing the solution will use a test based approach in order to ensure that the desired functionality is achieved.
The programming language will be Java, explicitly the versions 8 and 11. Both students use IntelliJ from JetBrains as their IDEA.
All of the code will be published under a ASF License.
The team will use github in order to synchronize its code bases. New features will be developed on their own branches.
Finally, the status of the project will be tracked using a board on Trello.

3.1. State of the art

Apache CXF is an open-source framework that implements the JAX-WS APIs which make building web services easy. JAX-WS encompasses many different areas.

CXF works with many different transports. Currently CXF includes support for HTTP, JMS, and Local (that is, "in-JVM") transports. The local transport is unique in that it will not work across machines, but simply sends messages in memory.

CXF enables the development of RESTful services via annotations using the HTTP Binding. By Using URI templates and annotations one binds a service operation to arbitrary URL/verb combinations. CXF will then listen for GET requests on that URL and using the parameter as a parameter to the service.

It is important to denote that, as CXF is working in web environment, it is essential to run on a web server in a background. In context of Java environments, a web-server is typically a HTTP Server running within a JVM application instance. Some examples of existing Java Web-servers are: Tomcat, Glassfish, Jetty, Netty, Undertow, Wildfly, WebLogic etc.

Applicable to a task given, there should be developer an http transport layer for Tomcat Web Server. Tomcat is an open source project guided and contributed by some enterprise companies. One of the reasons Tomcat became so popular among developers is due to its simplicity, maintainability and support.

Implementing the task means not only to investigate current Tomcat internal working model, but to make CXF run into an embedded Tomcat server. By embedded in this context will be understood that the server instance is located inside of .jar or .war application file.

Not everything will be done from scratch, of course. Of the few web servers that have already been implemented, there are especially Jetty and Undertow, which will be of interest for us as they are quite similar to Tomcat. Hence, we intend to study these in order to provide a solution to embed Tomcat as a transport in CXF.

4. Constraints and elements of risk

The biggest risk of this project is underrating the required time for the project. This is due to the fact that several different technologies have to be well understood before being able to propose a solution. Therefore, the time will be rather largely distributed to the different tasks in order to provide enough margin.

Another risk is that an unequal amount of work could result between the members of the team: Since the complexity of certain tasks is hard to predict and measure, one team member might spend a lot more time on a task than expected. This results in one team member spending much more time on the project than the other.

While both team members have experience using Java, they did not use Tomcat or CXF to their full extent. Hence, a major risk is missing knowledge in these two technologies. Especially Tomcat should be well understood, as necessary changes should (ideally) be applied to it instead of CXF.

Furthermore, both projects specify minimal requirements regarding performance. These must be achieved by the provided solution, too. Red Hat has a tool for checking the performance, which will also be used to assess the performance impact of the developed implementation.

As the upstream projects are in active development, it might be required to adapt to their changes. This requires a certain amount of flexibility and continuous observation of the upstream project in order to quickly adapt to a new version.

There is also a gap in knowledge regarding the web servers Jetty and Undertow. These should be understood in order to understand the necessary changes required for Tomcat. This will also help estimating the time more accurately, hence tackling the first risk.

If all the requirements are fulfilled in order to use tomcat as a transport in CXF, it’s important that the coding guidelines, test coverage and documentation imposed by Tomcat/Apache have to be followed. Otherwise, a refactoring might be required before merging the solution into the official github projects.

5. Deliverable goods

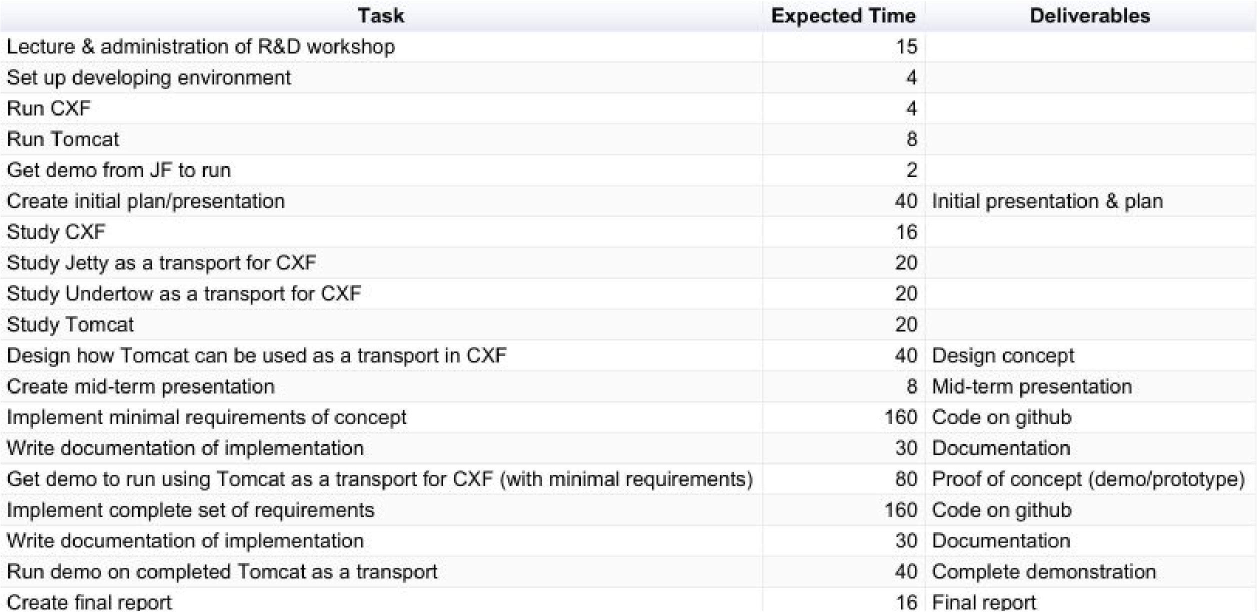
The following items will be delivered at the end of the project to the client:

- The code written in order to use Tomcat as a transport for CXF has to be made available to the client online on github. The correct functioning of the code should be ensured by writing unit tests. These will also be made available on github as part of the repository. Link to the repository: https://github.com/vaano94/TomcatCXF
- The complete documentation, especially the proposed concept including its state will be made available publicly on the same github repository. This will also include the current state of the project and what further work is required.
- The proof of concept in form of a prototype or demonstration will be delivered at the end of the project to the client.
- (Optional) If the code is complete and working properly, it should be merged with the official Apache CXF and Tomcat github project.

6. Efforts and schedule

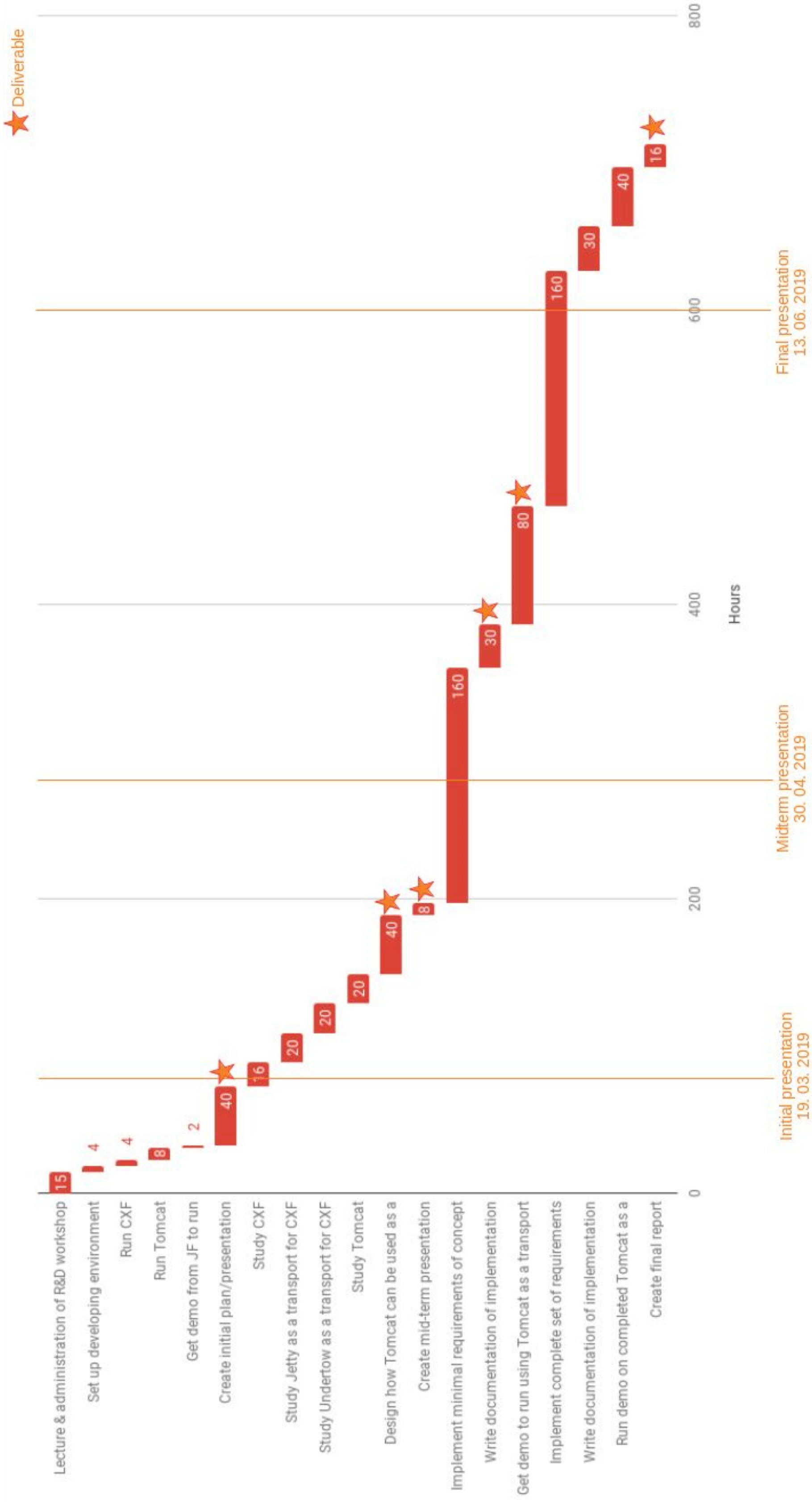
As mentioned before the main goal of this project is to deliver a proof of concept, which shows that an embedded Tomcat can be used as a transport in CXF. Due to the difficulty regarding time estimations, it's currently expected that the proof of concept will be realized after approximately 450 hours. Hence, there is a margin of 150 hours to reach the expected 600 hours that the team should use for the project. If the margin is not used, the time left (approximately 150 hours) will be used to complete the implementation. This would provide a fully functional embedded Tomcat as a transport in CXF.

The proof of concept and the fully functional implementation will be tested by Red Hat to ensure that the minimal expectations for CXF and Tomcat are met. This means for example that the provided solution should neither require a lot of additional computational resources nor take a lot of time to respond to requests. This ensures that a solution will be provided with a minimal standard of quality.



Task	Expected Time	Deliverables
Lecture & administration of R&D workshop	15	
Set up developing environment	4	
Run CXF	4	
Run Tomcat	8	
Get demo from JF to run	2	
Create initial plan/presentation	40	Initial presentation & plan
Study CXF	16	
Study Jetty as a transport for CXF	20	
Study Undertow as a transport for CXF	20	
Study Tomcat	20	
Design how Tomcat can be used as a transport in CXF	40	Design concept
Create mid-term presentation	8	Mid-term presentation
Implement minimal requirements of concept	160	Code on github
Write documentation of implementation	30	Documentation
Get demo to run using Tomcat as a transport for CXF (with minimal requirements)	80	Proof of concept (demo/prototype)
Implement complete set of requirements	160	Code on github
Write documentation of implementation	30	Documentation
Run demo on completed Tomcat as a transport	40	Complete demonstration
Create final report	16	Final report

Tomcat as a transport in CXF


Deliverable
Lecture & administration of R&D workshop
Set up developing environment
Run CXF
Run Tomcat
Get demo from JF to run
Create initial plan/presentation
Study CXF
Study Jetty as a transport for CXF
Study Undertow as a transport for CXF
Study Tomcat
Design how Tomcat can be used as a
Create mid-term presentation
Implement minimal requirements of concept
Write documentation of implementation
Get demo to run using Tomcat as a transport
Implement complete set of requirements
Write documentation of implementation
Run demo on completed Tomcat as a
Create final report
Hours
Initial presentation
Midterm presentation
Final presentation