CastafioreProposal

Abstract

Castafiore framework is a web frameworks, fully component oriented, ajax based, one page appplication. Although the framework can be viewed as a classical component oriented web framework like GWT, it has been designed and implemented with a particular goal in mind. It is to be able to make existing javascript libraries "java-able". Meaning that the framework allows to easily take a javascript library like e.g jquery ui and use the components and features via java. The integration of the javascript libraries need to be easy and natural.

Actually companies like ext js and smartgwt have made their components "java-able" using GWT. I wish to provide an alternative to gwt that is much easier, fast, lightweight, and much more productive.

Proposal

Castafiore framework has been designed and implemented with the following goals in mind

1. Fully object oriented
2. Simple API that is very close to html itself

We wanted an API that is closer to html markup and javascripts event model. I believe that this would help web developers easily visualize the rendering when reading source code.

1. Write a full application with only java
2. No need for compiler to convert to javascript like gwt.Javascript is rendered at runtime.
3. The same API can be used to make an software that is server centric and or client centric. Meaning that the user can make part of his application server centric and part of his application client centric. He can of course choose to make the whole application client centric or server centric depending on the requirement. All of this using the same API.
4. Low memory footprint
5. Easy packaging of application. Everything can be packaged in a single jar. This includes images, css or javascript as well
6. Same API used to create custom components. No need to external set of API or specific programming technique to create custom components. Creating a custom components should be done only the same way as writing an application.
7. No need for javascript to create custom components.
8. Although there is no need for javascript to create an application, it should be easy to integrate external javascript libraries and use them in java codes itself. This should be done is a natural way just like we would do in an HTML page. This should be like this so that javascript library authors with some java skills find it natural and easy to provide a castafiore component together with the library. e.g. The author of flexgrid should should find it easy to create a castafiore component thus making his library usable directly in java. Furthermore, with the fact that everything can be packaged in a single jar, a programmer, will simply need to download the jar, add it in his classpath, instantiate the necessary classes to be able to use a grid in his application.

Actually, we wanted to create java web framework for javascript programmers. They should find it easy and fun to integrate and distribute their js libraries as a castafiore component. They will be able to distribute their libraries as a single jar and this is very convenient for java developers to just download the jar, include in classpath, and using the library, components etc right away in their web application.

1. Load external resources like css and javascript lazily and efficiently only when needed.
2. Loading of external resources can be done eagerly if the API user wish so.
3. The framework does not own the whole page. Meaning that an application created with the framework can be used in an already created page. This allows API developer to use the framework only for very specific purposes like for example a dynamic table, while at the same time using other web framework like struts for the other sections of the application.
4. Although a whole application can be created using pure java, the framework should be able to integrate templates.
5. By default, there should be 100% separation of logic and presentation. Meaning that the API user should be able to take an html template, and dynamise it without needing to add any modification in the template.
6. On the other hand, if an API user is more script centric, he is able to write groovy template, jstl templates or any other template engine.
7. The API user should be able to easily use any template engine he wishes easily and naturally.
8. Event model is in 3 phase. Client-> server->back to client. API user should be able to write events in java that starts executing on the browser, then goes on the server if he wishes, then continue executing on the browser. e.g. We should be able to add an event on a button that starts by executing some javascript codes, then take some parameters and execute codes on the server, then continue back with some parameter from the server to execute some more javascript codes on the client. This technique if used skillfully helps to create very smooth application that constantly communicates with the server without any flicker
9. Support for websocket when the browser supports it.
10. Support for HTML5
11. Same API to write SVG applications.
12. Same API to write html5 canva application.
13. Support for browser back button
14. Crawlable by google crawler
15. Ultimately, the framework will come with a very rich set of re-usable components
16. Handle browser quirks
17. Built in validation

Background

The initial development of castafiore framework started back 3years ago by Kureem Rossaye. Later it was mature enough to be used in production. A whole web based platform was created. It consists of applications like Inventory system, CMS, DMS, account system, organization management system and also an advanced WYSIWYG online application builder. Many websites ranging from simple catalog website to advanced e-commerce website were also created using the framework.

Now I wish to donate this framework to the ASF in the hope to build a community and see it strive

Rationale

While there exists many web frameworks, there are very few that address all the requirements at the same time. There is a need for a web applications that at the same time can free user from javascript, html and css and at the same time easily integrate them when needed. There is also the need for a web framework that at the same time can be server centric and client centric. For example, the front-end of a web site should be made client centric and the back end server centric. While the whole application can be written in java, the web framework allows to create a whole application using traditional methods like html css javascript and page oriented framework like struts or spring mvc and at the same time implement only very specific parts using the framework.

Today many javascript library providers and making their libraries “java-able”. e.g. SmartGWT or extjs

They are using GWT for this purpose. Writing custom components with GWT can be extremely complex, error prone, difficult to debug and stabilize. Not mentioning the compilation nightmare. And GWT is exclusively client centric. Connection with the server is done exclusively via web services. On the other hand, castafiore framework helps to make javascript libraries “java-able” very easily. It is easy to debug and stabilize components since there is no such thing as compiler to generate javascript.

The framework is built upon jquery. Indeed jquery is an excellent javascript library. The framework basically acts as a thin layer over jquery. The result is a very simple and intuitive api which helps to write highly dynamic applications with lesser code. Just like jquery.

The 3 phase event model helps to bring the server very close to the browser. Meaning that with the same API, it is easy to interact with the browser and interact with the server. This helps to create very smooth running applications and at the same time interacting with the server and progressively render new components.

Thanks to the framework we will be able to harness the new HTML 5 api like websocket, workers, storage system, canvas and others using pure java. This is pretty exciting.

And hopefully, thanks to the ease at which we can integrate js libraries as a castafiore components, we will be able to use already mature js libraries in our java codes and interact with the server easily

Current Status

The rendering engine of the framework is completed and is being used in production. A large set of components are also being used. The javadocs and documentations are 90% completed. However there are many components that need to be stabilized and of course, there is the need to create more interesting, useful and sexy components for the pleasure of every programmer.

We need to create a good set of demo and sample codes

There is also a need to create a website where contributors can publish their custom components together with their license and demo

I intend to donate the WYSIWYG application builder to the ASF and host it online. This will be used as a sandbox or even can be used to create custom components and publish them online itself, without the need to download or install anything. I believe that this will help javascript library authors to make their libraries “java-able”. To attain this objective, we need to stabilize and increase security level on the application.

Meritocracy

- By submitting this incubator proposal, we’re expressing our intent to build a diverse developer community around Castafiore that will conduct itself according to The Apache Way and use meritocratic means of accepting contributions.

Community

Castafiore is actively being developed and maintained within Archnet Ltd by myself. By open sourcing it, I hope to attract contributors and build a vibrant community around the project. I will do my best to provide all necessary documentation and helps as swiftly as possible to help anybody interested in contributing on the project.

Core Developers

Castafiore is currently being developed by only Kureem Rossaye

Alignment

The ASF is a natural choice to host the Castafiore project, given the goal of open sourcing the project and fostering a community to grow and support the software.

Castafiore is a web framework tested and developed on apache-tomcat and websocket features uses implementation provided by tomcat7

Known Risks

Orphaned Products

There is currently one single developer for this project. However, this is the main reason why I want to host this project to ASF in order to create a community in order make the project less dependent on only myself. I also express my intention to provide all necessary help and documentation to help those interested to get started with the project.

Every aspect of the framework will be heavily documented to give clear understanding of the rendering engine in order to help anybody modify, enhance it.

Furthermore, the framework is being used in production in my company Archnet ltd and has been deployed in several companies in Mauritius.

Inexperience with Open Source

Indeed, I have low level of experience in open source and has never contributed in any opensource, although I have always wanted to do so. However, by working with our mentor and the Apache community I believe I will be able to conduct myself in accordance with the Apache Incubator guidelines.

Homogenous Developers

Actually there is only myself who is developing this project. This is actually one of the reasons why I want to submit this proposal. In order to be able to attract developers from various countries and companies. As I have already mentioned, I will be dedicated to provide everything required to help get started with contributing on this project.

Reliance on Salaried Developers

I am the sole developer of this project. I am the director of my company, Archnet ltd. So there is no reliance on salaried developers for this project.

Relationships with Other Apache Products

The project is a web frameworks that has been developed and tested on apache tomcat. Furthermore, the websocket feature actually uses exclusively the implementation provided by apache 7

An Excessive Fascination with the Apache Brand

- While we respect the reputation of the Apache brand and have no doubts that it will attract contributors and users, our interest is primarily to give Castafiore a solid home as an open source project following an established development model. We have also given reasons in the Rationale and Alignment sections.

Documentation

Project documentation exists and will be update online soon

Initial Source

https://github.com/archnetltd/castafioreframework/tree/trunk

Source and Intellectual Property Submission Plan

All the codes are available online on github. https://github.com/archnetltd/castafioreframework/tree/trunk

Since I am actually the only developer on this project, it is very easy for me to change the license to whatever license that best suits the ASF.

External Dependencies

Castafiore depends on spring framework and commons-fileupload. They are all Apache compatible.

Cryptography

Not applicable.

Required Resources

Mailing Lists

- castafiore-private for private PMC discussions
- castafiore-dev
- castafiore-commits
- castafiore-user
- castafiore-issues

Subversion Directory

- We prefer to use Git as our source control system: git://git.apache.org/castafiore

Issue Tracking

- JIRA Castafiore

Initial Committers

- Kureem Rossaye (kureem at gmail dot com)

Affiliations

- Archnetltd

Interested Parties

*

Sponsors

Champion

Nominated Mentors

*

Sponsoring Entity

Incubator PMC