

HomeTutorialsJoin UsAboutResourcesJob BoardExamplesWhitepapersAcademy



# Java Code Geeks

JAVA 2 JAVA DEVELOPERS RESOURCE CENTER

JavaAndroidJVM LanguagesSoftware DevelopmentAgileDevOpsCommunicationsCareerMiscMeta JCG

You are here: [Home](#) > [Enterprise Java](#) > CometD: Facebook similar chat for your Java web application

January 6, 5687 2:20 pm



**About Semika Kaluge**

## CometD: Facebook similar chat for your Java web application

by Semika Kaluge on October 17th, 2012 | Filed In: [Enterprise Java](#) Tags: [CometD](#), [Facebook](#)

Chatting is easy just like eating a piece of cake or drinking a hot coffee. Have you ever thought about developing a chat program by yourself?. You know that, it is not easy as chatting. But, if you are a developer and if you read to the end of this article, you may put a try to develop a chatting application by your self and allow your users to chat via your web application. I had to implement a chatting application for my web application. As every one does, I started to search on internet. I found IRC. When I read and search more about IRC, I understood that finding a web base client for IRC was difficult. I wanted to have more customizable web client which is working similar to Facebook. At last and luckily, I found CometD.

Finally, I was able to implement chatting application by using [CometD](#) and more customizable chat windows opening on the browser which is exactly similar to Facebook. This works almost all the modern browsers. This article explains step by step, How to implement chatting application from the scratch and also How to integrate chatting application to your existing Java base web application. Remember, Your web application should be a Java base one.

You need to [download the cometD](#) from their official web site. It has all the dependencies required to implement the chatting application except two javascript libraries. I have written two Javascript libraries, one to create dynamic chat windows like Facebook and other to handle CometD chatting functionality in generic way. If you can manage these stuff by your self, you don't need to use those two Javascript libraries. Actually, CometD documentation provides good details. But, I go ahead with the tutorial by using those two libraries. Any way, I recommend first use those two libraries and then customize it as you need. I hope to share the sample application with you and you can deploy it in your localhost and test, how it works.

### 1.Adding required jar files.

If you use maven to build your project, add the following dependencies into your pom.xml file

```
01 <dependencies>
02   <dependency>
03     <groupId>org.cometd.java</groupId>
04     <artifactId>bayeux-api</artifactId>
05     <version>2.5.0</version>
06   </dependency>
07   <dependency>
08     <groupId>org.cometd.java</groupId>
09     <artifactId>cometd-java-server</artifactId>
10     <version>2.5.0</version>
11   </dependency>
12   <dependency>
13     <groupId>org.cometd.java</groupId>
14     <artifactId>cometd-websocket-jetty</artifactId>
15     <version>2.5.0</version>
16     <exclusions>
17       <exclusion>
18         <groupId>org.cometd.java</groupId>
19         <artifactId>cometd-java-client</artifactId>
20       </exclusion>
21     </exclusions>
22   </dependency>
23   <dependency>
24     <groupId>org.slf4j</groupId>
25     <artifactId>slf4j-simple</artifactId>
26     <version>1.6.6</version>
27   </dependency>
28   <dependency>
29     <groupId>org.cometd.java</groupId>
30     <artifactId>cometd-java-annotations</artifactId>
31     <version>2.5.0</version>
32   </dependency>
33 </dependencies>
```

If you are not using maven to build your project, just copy the following .jar files into `/WEB-INF/lib` folder from your CometD download bundle. You can find these .jar files from `/cometd-demo/target/cometd-demo-2.5.0.war` file.

- bayeux-api-2.5.0.jar
- cometd-java-annotations-2.5.0.jar

**New sletter**

**67835** insiders are already enjoying weekly updates and complimentary whitepapers!

**Join them now** to gain [exclusive access](#) to the latest news in the Java world, as well as insights about Android, Scala, Groovy and other related technologies.

**Email address:**

**Join Us**



With **1,043,221** monthly unique visitors and over **500** authors we are placed among the top Java related sites around. Constantly being on the lookout for partners; we encourage you to join us. So if you have a blog with unique and interesting content then you should check out our [JCG partners program](#). You can also be a [guest writer](#) for Java Code Geeks and hone your writing skills!

**Recent Jobs**

[Java Applications Developer](#)  
Springfield, VA

[Java Developer / Systems Integration Developer](#)  
Washington, DC

[Senior Software Engineer - Java and Python](#)  
Raleigh, NC

[JAVA Developer](#)

1 trong 7

1/6/2015 2:20 PM

- cometd-java-common-2.5.0.jar
- cometd-java-server-2.5.0.jar
- cometd-w ebsocket-jetty-2.5.0.jar
- javax.inject-1.jar
- jetty-continuation-7.6.7.v20120910.jar
- jetty-http-7.6.7.v20120910.jar
- jetty-io-7.6.7.v20120910.jar
- jetty-jmx-7.6.7.v20120910.jar
- jetty-util-7.6.7.v20120910.jar
- jetty-w ebsocket-7.6.7.v20120910.jar
- jsr250-api-1.0.jar
- slf4j-api-1.6.6.jar
- slf4j-simple-1.6.6.jar

Reston, VA  
Software Engineer (JAVA)  
Reston, VA

[View All](#)

## 2.Adding required Javascript files.

You need to link the following Javascript files.

- cometd.js
- AckExtension.js
- ReloadExtension.js
- jquery-1.8.2.js
- jquery.cookie.js
- jquery.cometd.js
- jquery.cometd-reload.js
- chat.window.js
- comet.chat.js

The 'chat.window.js' and 'comet.chat.js' are my own two Javascript libraries which does not come with CometD distribution. If you are totally following this tutorial, you have to link those two libraries as well. Provided sample application has these two Javascript libraries.

## 3.Writing chat service class.

```
001 /**
002  * @author Semika siriwardana
003  * CometD chat service.
004  */
005 package com.semika.cometd;
006
007 import java.util.HashMap;
008 import java.util.Map;
009 import java.util.Set;
010 import java.util.concurrent.ConcurrentHashMap;
011 import java.util.concurrent.ConcurrentMap;
012
013 import javax.inject.Inject;
014
015 import org.cometd.annotation.Configure;
016 import org.cometd.annotation.Listener;
017 import org.cometd.annotation.Service;
018 import org.cometd.annotation.Session;
019 import org.cometd.bayeux.client.ClientSessionChannel;
020 import org.cometd.bayeux.server.BayeuxServer;
021 import org.cometd.bayeux.server.ConfigurableServerChannel;
022 import org.cometd.bayeux.server.ServerMessage;
023 import org.cometd.bayeux.server.ServerSession;
024 import org.cometd.server.authorizer.GrantAuthorizer;
025 import org.cometd.server.filter.DataFilter;
026 import org.cometd.server.filter.DataFilterMessageListener;
027 import org.cometd.server.filter.JSONDataFilter;
028 import org.cometd.server.filter.NoMarkupFilter;
029
030 @Service('chat')
031 public class ChatService {
032     private final ConcurrentMap<String, Map<String, String>> _members = new
033     ConcurrentHashMap<String, Map<String, String>>();
034     @Inject
035     private BayeuxServer _bayeux;
036     @Session
037     private ServerSession _session;
038
039     @Configure({'/chat/**', '/members/**'})
040     protected void configureChatStarStar(ConfigurableServerChannel channel) {
041         DataFilterMessageListener noMarkup = new DataFilterMessageListener(new NoMarkupFilter(), new
042         BadWordFilter());
043         channel.addListener(noMarkup);
044         channel.addAuthorizer(GrantAuthorizer.GRANT_ALL);
045     }
046
047     @Configure('/service/members')
048     protected void configureMembers(ConfigurableServerChannel channel) {
049         channel.addAuthorizer(GrantAuthorizer.GRANT_PUBLISH);
050         channel.setPersistent(true);
051     }
052
053     @Listener('/service/members')
054     public void handleMembership(ServerSession client, ServerMessage message) {
055         Map<String, Object> data = message.getDataAsMap();
056         final String room = ((String) data.get('room')).substring('/chat/'.length());
057
058         Map<String, String> roomMembers = _members.get(room);
059         if (roomMembers == null) {
060             Map<String, String> new_room = new ConcurrentHashMap<String, String>();
061         }
062     }
063 }
```



```

059         roomMembers = _members.putIfAbsent(room, new_room);
060         if (roomMembers == null) roomMembers = new_room;
061     }
062     final Map<String, String> members = roomMembers;
063     String userName = (String)data.get('user');
064     members.put(userName, client.getId());
065     client.addListener(new ServerSession.RemoveListener() {
066         public void removed(ServerSession session, boolean timeout) {
067             members.values().remove(session.getId());
068             broadcastMembers(room, members.keySet());
069         }
070     });
071
072     broadcastMembers(room, members.keySet());
073 }
074
075 private void broadcastMembers(String room, Set<String> members) {
076     // Broadcast the new members list
077     ClientSessionChannel channel = _session.getLocalSession().getChannel('/members/' + room);
078     channel.publish(members);
079 }
080
081 @Configure('/service/privatechat')
082 protected void configurePrivateChat(ConfigurableServerChannel channel) {
083     DataFilterMessageListener noMarkup = new DataFilterMessageListener(new NoMarkupFilter(), new
BadWordFilter());
084     channel.setPersistent(true);
085     channel.addListener(noMarkup);
086     channel.addAuthorizer(GrantAuthorizer.GRANT_PUBLISH);
087 }
088
089 @Listener('/service/privatechat')
090 protected void privateChat(ServerSession client, ServerMessage message) {
091     Map<String, Object> data = message.getDataAsMap();
092
093     String room = ((String)data.get('room')).substring('/chat/'.length());
094     Map<String, String> membersMap = _members.get(room);
095     if (membersMap == null) {
096         Map<String, String> new_room = new ConcurrentHashMap<String, String>();
097         membersMap = _members.putIfAbsent(room, new_room);
098         if (membersMap == null)
099             membersMap = new_room;
100     }
101
102     String peerName = (String)data.get('peer');
103     String peerId = membersMap.get(peerName);
104
105     if (peerId != null) {
106         ServerSession peer = _bayeux.getSession(peerId);
107
108         if (peer != null) {
109             Map<String, Object> chat = new HashMap<String, Object>();
110             String text = (String)data.get('chat');
111             chat.put('chat', text);
112             chat.put('user', data.get('user'));
113             chat.put('scope', 'private');
114             chat.put('peer', peerName);
115             ServerMessage.Mutable forward = _bayeux.newMessage();
116             forward.setChannel('/chat/' + room);
117             forward.setId(message.getId());
118             forward.setData(chat);
119
120             if (text.lastIndexOf('lazy') > 0) {
121                 forward.setLazy(true);
122             }
123             if (peer != client) {
124                 peer.deliver(_session, forward);
125             }
126             client.deliver(_session, forward);
127         }
128     }
129 }
130
131 }
132
133 class BadWordFilter extends JSONDataFilter {
134     @Override
135     protected Object filterString(String string) {
136         if (string.indexOf('dang') >= 0) {
137             throw new DataFilter.Abort();
138         }
139         return string;
140     }
141 }

```

#### 4. Changing web.xml file.

You should add the following filter into your web.xml file.

```

1 <filter>
2   <filter-name>continuation</filter-name>
3   <filter-class>org.eclipse.jetty.continuation.ContinuationFilter</filter-class>
4 </filter>
5 <filter-mapping>
6   <filter-name>continuation</filter-name>
7   <url-pattern>/cometd/*</url-pattern>
8 </filter-mapping>

```

And also the following servlet.

```

01 <servlet>
02   <servlet-name>cometd</servlet-name>
03   <servlet-class>org.cometd.annotation.AnnotationCometdServlet</servlet-class>
04   <init-param>
05     <param-name>timeout</param-name>
06     <param-value>20000</param-value>

```



```
07 </init-param>
08 <init-param>
09   <param-name>interval</param-name>
10   <param-value>0</param-value>
11 </init-param>
12 <init-param>
13   <param-name>maxInterval</param-name>
14   <param-value>10000</param-value>
15 </init-param>
16 <init-param>
17   <param-name>maxLazyTimeout</param-name>
18   <param-value>5000</param-value>
19 </init-param>
20 <init-param>
21   <param-name>long-polling.multiSessionInterval</param-name>
22   <param-value>2000</param-value>
23 </init-param>
24 <init-param>
25   <param-name>logLevel</param-name>
26   <param-value>0</param-value>
27 </init-param>
28 <init-param>
29   <param-name>transports</param-name>
30   <param-value>org.cometd.websocket.server.WebSocketTransport</param-value>
31 </init-param>
32 <init-param>
33   <param-name>services</param-name>
34   <param-value>com.semika.cometd.ChatService</param-value>
35 </init-param>
36 <load-on-startup>1</load-on-startup>
37 </servlet>
38 <servlet-mapping>
39   <servlet-name>cometd</servlet-name>
40   <url-pattern>/cometd/*</url-pattern>
41 </servlet-mapping>
```

### 5.Implementing client side functions.

I think this section should be descriptive. If you allow s your users to chat w ith other users, you need to show the list of online users in you w eb page, just like Facebook show s the online users inside the right side bar. For that, you can place a simple <span> or <div> tag inside your page. I have done it as follow s.

```
1 <div id='members'></div>
```

All the online users w ill be displayed with in the above container. Once you click on a particular user name, it w ill open a new chat window similar to Facebook. For each pair of users, it w ill open a new chat window . To get this behaviour, you should use 'chat.window.js' w hich I mentioned before. Chatting in betw een particular pair of users w ill continue through a dedicated chat window .

Just after user is logging into your web application as usual way, we should subscribe that user to chat channels. You can do it using the follow ing way.

```
1 $(document).ready(function(){
2   $.cometChat.onLoad({memberListContainerID: 'members'});
3 });
```

Note that, I have passed the 'id' of online user list container as a configuration parameter. Then, user should be joined w ith channel as follow s.You can call the bellow method w ith the username.

```
1 function join(userName){
2   $.cometChat.join(userName);
3 }
```

Since for each chat, there is a dedicated chat window just like Facebook, we should maintain global Javascript array to store those created chat window objects. You need to place the follow ing Javascript code inside your page.

```
01 function getChatWindowByUserPair(loginUserName, peerUserName) {
02   var chatWindow;
03   for(var i = 0; i < chatWindowArray.length; i++) {
04     var windowInfo = chatWindowArray[i];
05     if (windowInfo.loginUserName == loginUserName && windowInfo.peerUserName == peerUserName) {
06       chatWindow = windowInfo.windowObj;
07     }
08   }
09   return chatWindow;
10 }
11
12 function createWindow(loginUserName, peerUserName) {
13   var chatWindow = getChatWindowByUserPair(loginUserName, peerUserName);
14   if (chatWindow == null) { //Not chat window created before for this user pair.
15     chatWindow = new ChatWindow(); //Create new chat window.
16     chatWindow.initWindow({
17       loginUserName:loginUserName,
18       peerUserName:peerUserName,
19       windowArray:chatWindowArray});
20
21     //collect all chat windows opened so far.
22     var chatWindowInfo = { peerUserName:peerUserName,
23       loginUserName:loginUserName,
24       windowObj:chatWindow
25     };
26
27     chatWindowArray.push(chatWindowInfo);
28   }
29   chatWindow.show();
30   return chatWindow;
31 }
```

As I mentioned above, declare follow ing global Javascript variable.



```

1 var chatWindowArray = [];
2 var config = {
3     contextPath: '${pageContext.request.contextPath}'
4 };

```

Since I am using a JSP page, I have to get the context path via '**pageContext**' variable. If you are using a HTML page, manage it by your self to declare 'config' Javascript global variable. Now, you almost reached to last part of the tutorial.

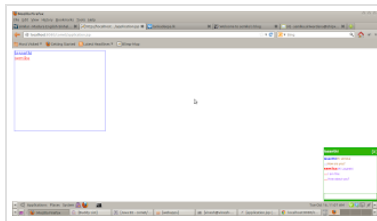
#### 5.How does the sample application works?

You can download the [comet.war](#) file and deploy it in your server. Point the browser to following URL.

**http://localhost:8080/comet**

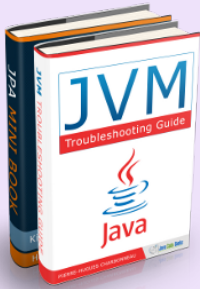
This will bring you to a page which has a text field and button called 'Join'. Insert some user name as you wish and click on 'Join' button. Then you will be forwarded to another page which has list of online users. Your name is highlighted in red color. To chat in your local machine, You can open another browser (IE and FF) and join to the chat channel. The peer user displays in blue color in the online users list. Once you click on a peer user, it will open a new chat window so that You can chat with him. This functions very similar to Facebook chatting.

I have tested this chatting application in IE, FF and Chrome and works fine. If you want any help of integrating this with your Java based web application, just send me a mail.



**Reference:** Facebook similar chat for your Java web application. from our JCG partner Semika loku kaluge at the [Code Box](#) blog.

### Do you want to know how to develop your skillset to become a **Java Rockstar**?



#### Subscribe to our newsletter to start Rocking right now!

To get you started we give you two of our best selling eBooks for **FREE**

##### JPA Mini Book

Learn how to leverage the power of JPA in order to create robust and flexible Java applications. With this Mini Book, you will get introduced to JPA and smoothly transition to more advanced concepts.

##### JVM Troubleshooting Guide

The Java virtual machine is really the foundation of any Java EE platform. Learn how to master it with this advanced guide!

Email address:

**Sign up!**

### 25 Responses to "CometD: Facebook similar chat for your Java web application"



**sathish**

November 10th, 2014 at 12:08 pm

hi,

I need your help. I downloaded comet.war and deployed it into eclipse. It was deployed successfully and then I ran the project. It displays the index page but when I click the join button, an error occurred. The error mentioned below:  
 101 [main] WARN org.cometd.annotation.AnnotationCometdServlet - Failed to create annotated service com.semika.cometd.ChatService



```
java.lang.ClassNotFoundException: com.semika.cometd.ChatService
at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1387)
at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1233)
at org.eclipse.jetty.util.Loader.loadClass(Loader.java:101)
at org.eclipse.jetty.util.Loader.loadClass(Loader.java:80)
at org.cometd.annotation.AnnotationCometdServlet.new Service(AnnotationCometdServlet.java:94)
at org.cometd.annotation.AnnotationCometdServlet.processService(AnnotationCometdServlet.java:80)
at org.cometd.annotation.AnnotationCometdServlet.init(AnnotationCometdServlet.java:64)
at javax.servlet.GenericServlet.init(GenericServlet.java:212)
at org.apache.catalina.core.StandardWrapper.loadServlet(StandardWrapper.java:1173)
at org.apache.catalina.core.StandardWrapper.load(StandardWrapper.java:993)
at org.apache.catalina.core.StandardContext.loadOnStartup(StandardContext.java:4149)
at org.apache.catalina.core.StandardContext.start(StandardContext.java:4458)
at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1045)
at org.apache.catalina.core.StandardHost.start(StandardHost.java:722)
at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1045)
at org.apache.catalina.core.StandardEngine.start(StandardEngine.java:443)
at org.apache.catalina.core.StandardService.start(StandardService.java:516)
at org.apache.catalina.core.StandardServer.start(StandardServer.java:710)
at org.apache.catalina.startup.Catalina.start(Catalina.java:583)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
at java.lang.reflect.Method.invoke(Unknown Source)
at org.apache.catalina.startup.Bootstrap.start(Bootstrap.java:288)
at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:413)
Nov 10, 2014 3:27:14 PM org.apache.catalina.core.StandardContext loadOnStartup
SEVERE: Servlet /comet threw load() exception
java.lang.ClassNotFoundException: com.semika.cometd.ChatService
at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1387)
at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1233)
at org.eclipse.jetty.util.Loader.loadClass(Loader.java:101)
at org.eclipse.jetty.util.Loader.loadClass(Loader.java:80)
at org.cometd.annotation.AnnotationCometdServlet.new Service(AnnotationCometdServlet.java:94)
at org.cometd.annotation.AnnotationCometdServlet.processService(AnnotationCometdServlet.java:80)
at org.cometd.annotation.AnnotationCometdServlet.init(AnnotationCometdServlet.java:64)
at javax.servlet.GenericServlet.init(GenericServlet.java:212)
at org.apache.catalina.core.StandardWrapper.loadServlet(StandardWrapper.java:1173)
at org.apache.catalina.core.StandardWrapper.load(StandardWrapper.java:993)
at org.apache.catalina.core.StandardContext.loadOnStartup(StandardContext.java:4149)
at org.apache.catalina.core.StandardContext.start(StandardContext.java:4458)
at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1045)
at org.apache.catalina.core.StandardHost.start(StandardHost.java:722)
at org.apache.catalina.core.ContainerBase.start(ContainerBase.java:1045)
at org.apache.catalina.core.StandardEngine.start(StandardEngine.java:443)
at org.apache.catalina.core.StandardService.start(StandardService.java:516)
at org.apache.catalina.core.StandardServer.start(StandardServer.java:710)
at org.apache.catalina.startup.Catalina.start(Catalina.java:583)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
at java.lang.reflect.Method.invoke(Unknown Source)
at org.apache.catalina.startup.Bootstrap.start(Bootstrap.java:288)
at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:413)
thanks in advance
```

[Reply](#)**Kernel**

December 2nd, 2014 at 4:30 pm

Where is the source code for the application?

[Reply](#)[« Older Comments](#)

### Leave a Reply

<input type="text"/>	Name (Required)
<input type="text"/>	Mail (will not be published) (Required)
<input type="text"/>	Website



7 - two =

☐ Notify me of follow up comments via e-mail. You can also [subscribe](#) without commenting.

☒ Sign me up for the new sletter!

Submit Comment

#### Knowledge Base

- [Academy](#)
- [Examples](#)
- [Resources](#)
- [Tutorials](#)
- [Whitepapers](#)

#### Partners

- [Mkyong](#)

#### The Code Geeks Network

- [Java Code Geeks](#)
- [.NET Code Geeks](#)
- [Web Code Geeks](#)

#### Hall Of Fame

- ["Android Full Application Tutorial" series](#)
- [GWT 2 Spring 3 JPA 2 Hibernate 3.5 Tutorial](#)
- [Advantages and Disadvantages of Cloud Computing – Cloud computing pros and cons](#)
- [Android Google Maps Tutorial](#)
- [Android Location Based Services Application – GPS location](#)
- [11 Online Learning websites that you should check out](#)
- [Java Best Practices – Vector vs ArrayList vs HashSet](#)
- [Android JSON Parsing with Gson Tutorial](#)
- [Android Quick Preferences Tutorial](#)
- [Difference between Comparator and Comparable in Java](#)

#### About Java Code Geeks

JCGs (Java Code Geeks) is an independent online community focused on creating the ultimate Java to Java developers resource center; targeted at the technical architect, technical team lead (senior developer), project manager and junior developers alike. JCGs serve the Java, SOA, Agile and Telecom communities with daily new s w ritten by domain experts, articles, tutorials, review s, announcements, code snippets and open source projects.

Java Code Geeks and all content copyright © 2010-2015, Exelixis Media Ltd | [Terms of Use](#) | [Privacy Policy](#) | [Contact](#)  
All trademarks and registered trademarks appearing on Java Code Geeks are the property of their respective owners.  
Java is a trademark or registered trademark of Oracle Corporation in the United States and other countries.  
Java Code Geeks is not connected to Oracle Corporation and is not sponsored by Oracle Corporation.

