

# Pushing Data to the Browser with Comet

by Kevin Nilson

<http://www.javaclimber.com>

**AjaxWorld 2008**

<http://ajaxworld.com>

Wednesday October 22, 2008

9:50 - 10:35am

# Kevin Nilson

Kevin Nilson is currently a Principal Software Architect for a Large Financial Institute in Silicon Valley. He is the co-lead of the [Silicon Valley Web Developer JUG](#). He is also the co-lead of the [Silicon Valley Google Technology User Group](#).

Prior to his current work, Kevin had extensive industry software engineering and consulting experience with companies such as: Pfizer, Northrop Grumman, AT&T Corporation, etc. He also worked part-time as an adjunct professor at the College of San Mateo.

Kevin received a Bachelor's degree in Computer Science with a Minor in Mathematics from Southern Illinois University in Carbondale and a Masters degree in Computer Science from Southern Illinois University in Edwardsville. He is a Sun Certified Java Programmer and Sun Certified Web Component Developer.

You can learn more about Kevin on his website [JavaClimber.com](#).

# Ajax

What is Ajax?

- Client Browser calling webpage to get data to update a page
- Many great JavaScript Ajax Frameworks
- JSON often used

# Uses of Ajax

- Fill Drop-downs to support selecting: State, County, then City
- Drill-down in reporting
- verify form data
- adding route to map

# Ajax Limitations

Based on Client Request, not server state change.

- Chat Application
- Stock Updates
- Temperature Update
- Server Load, Server Processes, Current Sessions
- JMS ...

Polling is the best you can do with Ajax

# What is Comet

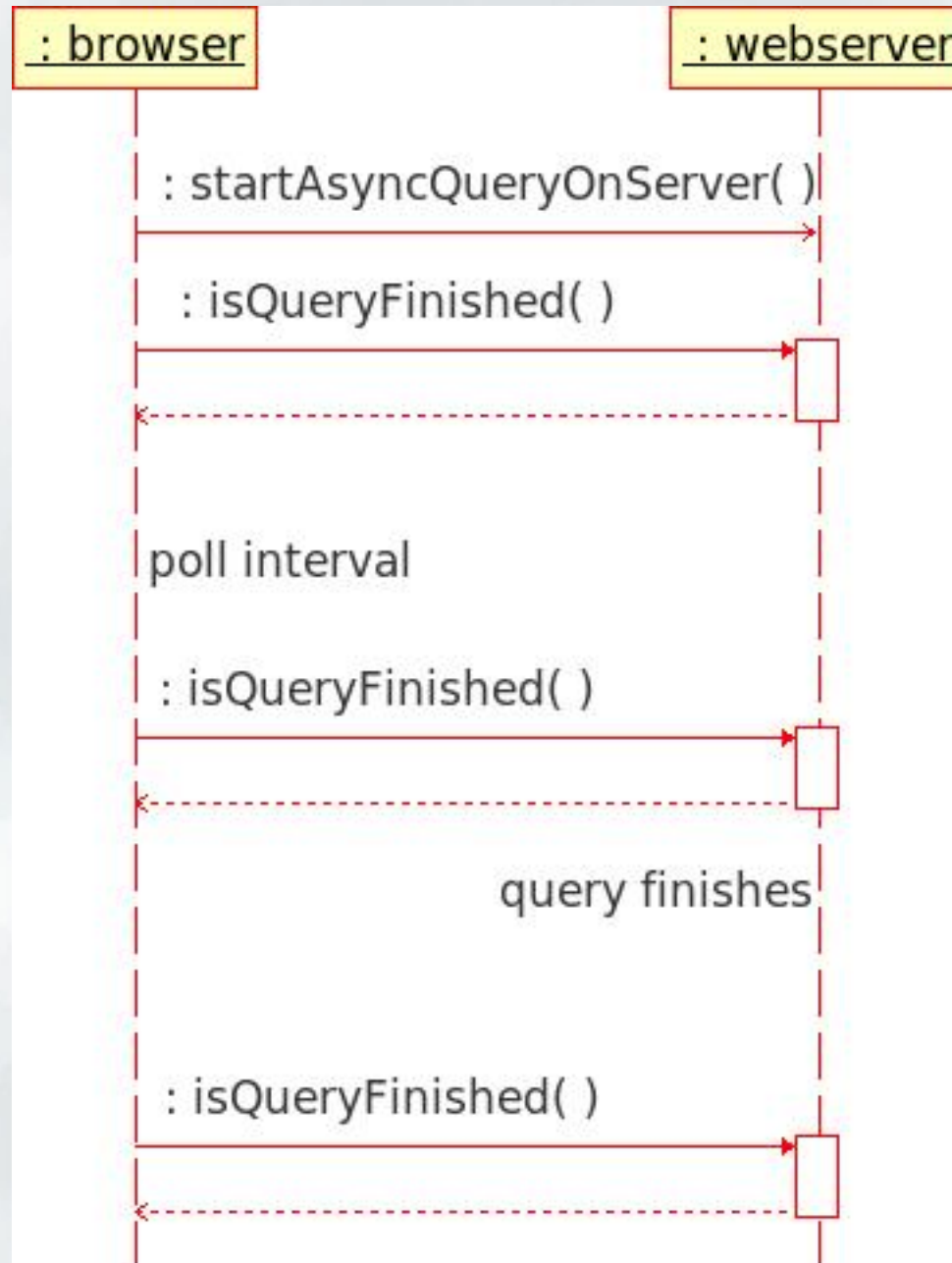
- Allows data to be pushed to the browser from the server
- Sometimes called Reverse Ajax
- Challenge because HTTP is a connectionless protocol
- Long Polling and Streaming are considered Comet (next)
- Push data to the browser without install
  - Requires accepting "scary" dialog, configuration, ....
  - Applet
  - Flash



# Uses of Comet

- Chat Application
- Stock Updates
- Temperature Update
- Server Load, Server Processes, Current Sessions
- JMS ...

# Polling (Option 1)



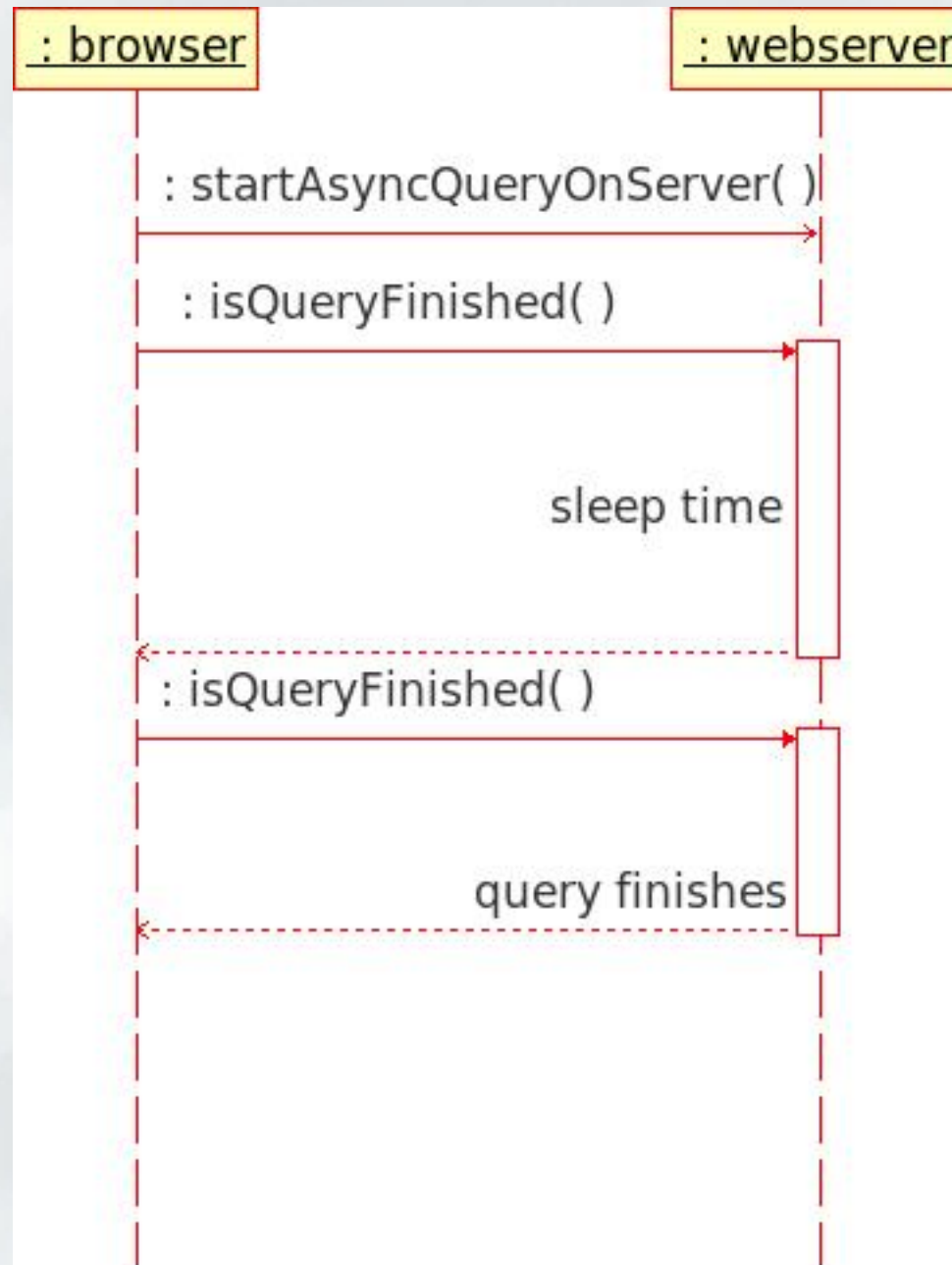


# Polling



# Polling Demo

# Long Polling (Option 2)





# Long Polling



# Long Polling - Chat

( [javaclimber.com](http://javaclimber.com) )

## JavaClimber.com - Kevin Nilson's Portal

[Home](#) [Information Technology](#) [Rock & Ice Climbing](#) **Chat** [Email](#) [Guestbook](#) [Configuration](#) [Links](#)

### About JavaClimber Chat

Build using GWT. Using Comet Long Polling.

### JavaClimber Chat

Users:

Kevin  
Duke

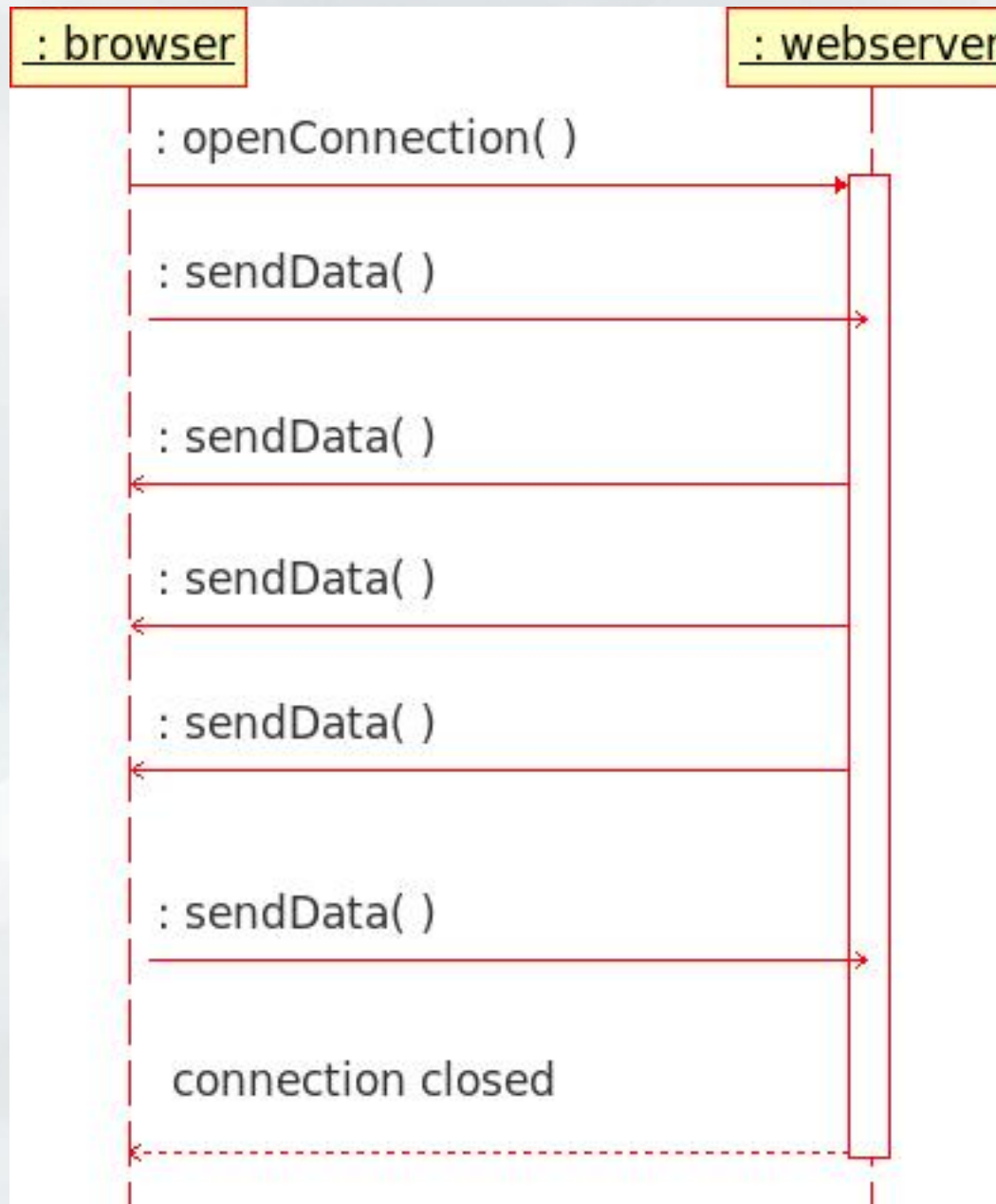
Chat:

Duke (9:33:33) ENTERED CHAT  
Duke (9:34:00) hello

# Long Polling Demo



# Streaming (Option 3)



# Streaming

What are all  
your favorite  
treats?



Peanuts are  
one of my  
favorites



Rice cakes  
are one of  
my  
favorites



Bones are  
my favorite  
treat!

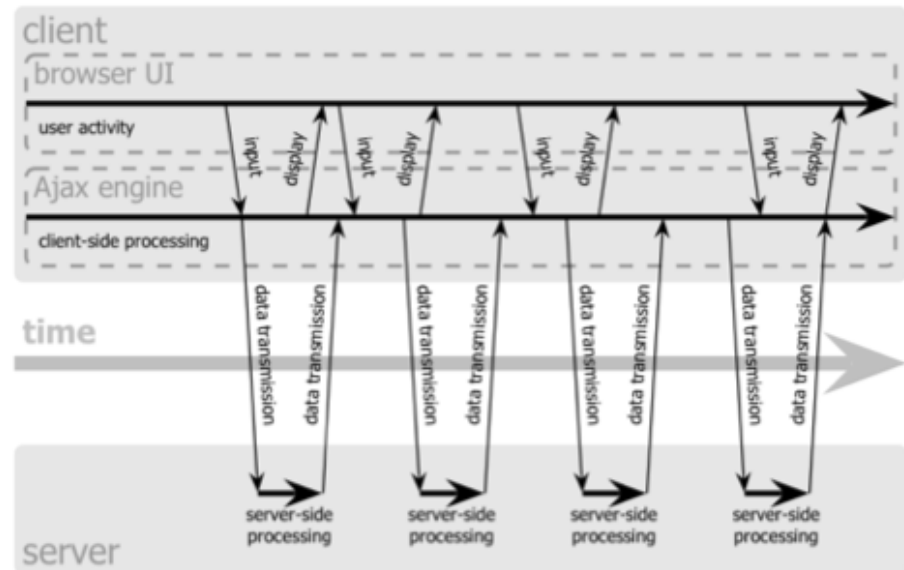


Ok, your  
favorites are  
peanuts, rice  
cakes, and  
bones!

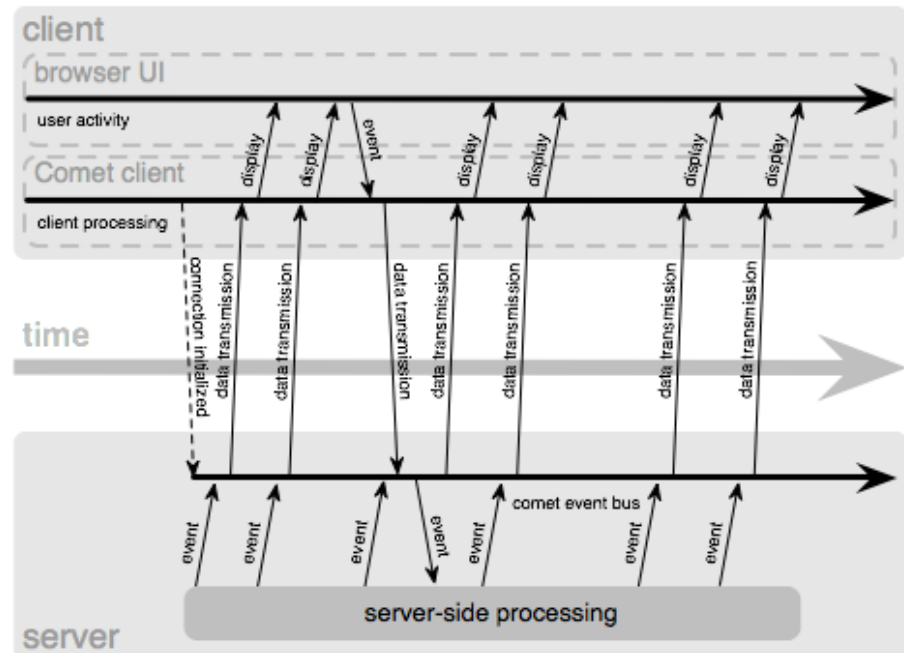


# Ajax vs Comet

Ajax web application model (asynchronous)



Comet web application model



# Challenges

## Client Side

- 2 request limit per domain

## Server Side

- Servlet = 1 thread per request

- Continuation added to Servlet 3

# Bayeux

- Only Open Standard for Comet, (Open Source)
- Publish / Subscribe
- In Draft
- Multiple Client and Server Implementations
  - Client
    - Dojo
    - JQuery
    - ExtJs
    - flexcomet ...
  - Server
    - Jetty
    - Tomcat
    - WebSphere
    - Weblogic
    - Cometd-twisted (python)
    - Cometd-perl

# Bayeux Sensor Demo



# Bayeux JavaScript - Publish

```
var text = 'hello';
```

```
dojox.cometd.publish("/chat", { chat: text});
```

# Bayeux JavaScript - Subscribe

```
dojox.cometd.init("http://localhost:8080/cometd");  
dojox.cometd.subscribe("/chat", myapp, "chatMessage");
```

```
chatMessage: function(message){  
    if(!message.data){  
        alert("bad chat data "+message);  
    } else {  
        updateChatDisplay(message.data.chat);  
    }  
}
```

# Bayeux - Reconnect

```
myapp._meta=dojo.subscribe("/cometd/meta",
    dojo.hitch(this,function(event){
        if (event.action=="handshake") {
        } else if (event.action=="connect") {
            if (event.successful && !this._connected){
                //user is reconnected
            }
            if (!event.successful && this._connected){
                //user is disconnected
            }
            this._connected=event.successful;
        }
    }));
```

# Bayeux - Java - Publish

```
public void publish(String channel, Map<String, Object> data) {  
    if(getBayeux().hasChannel(channel)){  
        Channel c = getBayeux().  
            getChannel(channel, false);  
        c.publish(getClient(), data,"fromserver");  
    }else{  
        //oops no subscribers  
    }  
}
```

# Bayeux - Java - Subscribe & Unsubscribe

```
super.subscribe(channel, method);
```

```
Channel c = bayeux.getChannel(channelStr,  
    true);  
c.unsubscribe(getClient());
```

# Bayeux - Java - Remove Notification

```
client.addListener(new RemoveListener( ){  
    public void removed(String clientId,  
        boolean timeout){  
  
        //do something to clean-up  
  
    }  
});
```



# Bayeux - Java - Security

```
public interface SecurityPolicy
{
    boolean canHandshake(Message message);
    boolean canCreate(Client client,String
        channel,Message message);
    boolean canSubscribe(Client client,String
        channel,Message message);
    boolean canPublish(Client client,String
        channel,Message message);
}
```

# Bayeux - Java - Request

```
AbstractBayeux b = (AbstractBayeux)
    getBayeux();
if (b.isRequestAvailable())
    HttpServletRequest request =
        b.getCurrentRequest();
```

web.xml, add to cometd servlet

```
<init-param>
  <param-name>requestAvailable</param-name>
  <param-value>true</param-value>
</init-param>
```

# Bayeux - Java - Request

```
AbstractBayeux b = (AbstractBayeux)
    getBayeux();
if (b.isRequestAvailable())
    HttpServletRequest request =
        b.getCurrentRequest();
```

web.xml, add to cometd servlet

```
<init-param>
  <param-name>requestAvailable</param-name>
  <param-value>true</param-value>
</init-param>
```

# Cometd Chat Demo

# Bayeux - Misconceptions

Many Believe Bayeux is only for Pub/Sub, but this is not true. Two special channels exist for RPC style requests

- meta channels
- service channels

# Future

HTML5 brings WebSockets

- Full Spec Completes 2022, but WebSockets spec is very mature
- Text based, so name is a bit confusing



# Questions

## *Article by Kevin Nilson*

**Pushing Data to the Browser with Comet** by Kevin Nilson

[http://www.developer.com/services/article.php/10928\\_3756841](http://www.developer.com/services/article.php/10928_3756841)

## *Upcoming Speaking Events with Kevin Nilson*

**The Chronicles of Web Standard: the HTML 5, the Comet and the WebSocket**

10/29/2008 - Silicon Valley Web Builder

<http://tinyurl.com/webstandard>

**An Introduction to Comet and Bayeux**

November 8-9, 2008 Silicon Valley Code Camp

<http://www.siliconvalley-codecamp.com>