HTML5 - SERVER SENT EVENTS

Conventional web applications generate events which are dispatched to the web server. For example a simple click on a link requests a new page from the server.

The type of events which are flowing from web browser to the web server may be called client-sent events.

Along with HTML5, WHATWG Web Applications 1.0 introduces events which flow from web server to the web browsers and they are called Server-Sent Events *SSE*. Using SSE you can push DOM events continously from your web server to the visitor's browser.

The event streaming approach opens a persistent connection to the server, sending data to the client when new information is available, eliminating the need for continuous polling.

Server-sent events standardizes how we stream data from the server to the client.

Web Application for SSE

To use Server-Sent Events in a web application, you would need to add an <eventsource> element to the document.

The **src** attribute of <eventsource> element should point to an URL which should provide a persistent HTTP connection that sends a data stream containing the events.

The URL would point to a PHP, PERL or any Python script which would take care of sending event data consistently. Following is a simple example of web application which would expect server time.

```
<!DOCTYPE HTML>
<html>
   <head>

      <script type="text/javascript">
         /* Define event handling logic here */
      </script>

   </head>
   <body>

      <div >
         <eventsource src="/cgi-bin/ticker.cgi" />
      </div>

      <div >
         <TIME>
      </div>

   </body>
</html>
```

Server Side Script for SSE

A server side script should send **Content-type** header specifying the type *text/event-stream* as follows.

```
print "Content-Type: text/event-stream\n\n";
```

After setting Content-Type, server side script would send an **Event:** tag followed by event name. Following example would send Server-Time as event name terminated by a new line character.

```
print "Event: server-time\n";
```

Final step is to send event data using **Data:** tag which would be followed by integer of string value terminated by a new line character as follows −

```
$time = localtime();
print "Data: $time\n";
```

Finally, following is complete ticker.cgi written in perl −

```
#!/usr/bin/perl

print "Content-Type: text/event-stream\n\n";

while(true){
   print "Event: server-time\n";
   $time = localtime();
   print "Data: $time\n";
   sleep(5);
}
```

Handle Server-Sent Events

Let us modify our web application to handle server-sent events. Following is the final example.

```
<!DOCTYPE HTML>
<html>
   <head>

      <script type="text/javascript">
         document.getElementsByTagName("eventsource")[0].addEventListener("server-time",
eventHandler, false);

         function eventHandler(event)
         {
            // Alert time sent by the server
            document.querySelector('#ticker').innerHTML = event.data;
         }
      </script>

   </head>
   <body>

      <div >
         <eventsource src="/cgi-bin/ticker.cgi" />
      </div>

      <div >
         [TIME]
      </div>

   </body>
</html>
```

Before testing Server-Sent events, I would suggest to make sure if your web browser supports this concept.