用Server Send Event实现服务端推送

jamie.wang 分享于 2013-04-17

由于在HTTP协议中，服务器是出于被动的位置，浏览器（客户端）发起HTTP请求从服务端获取数据。但web2.0时代的一些web应用，则需要及时更新服务端的信息，（例如：票务，航班信息，股票信息等），还有一些情况下需要客户端和服务端实现一种异步通信的机制（比如：客户端提交一个耗时的请求，服务端处理完成后，通知客户端）。  
  
  
HTML5提供了WebSocket，客户端和服务端可以建立一个双工的通信通道，相互发送数据。  
  
Server Send Event（SSE）是一种简单的浏览器实现的轮询，并可以指定事件类型，事件id等。  
  
而由于HTML5出于draft阶段，并且浏览器支持的程度相差较大，因此现在很多是通过客户端轮询获取结果的。  
  
有时候你的web应用的用户是较小范围可控的，用用HTML5的技术还是比较惬意的。  
  
  
下面是一个SSE的例子。  
  
服务端代码，一个简单的servlet，在获取请求后做一个很耗时的操作，然后把结果告知客户端。  
注意SSE的头：text/event-stream，和发送数据的格式：data:message\n\n

**package** org.jamee.demo.webapp;

**import** java.io.PrintWriter;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**public** **class** **SSEDemo** **extends** **HttpServlet** {

**private** **static** **final** **long** serialVersionUID = 6205864980814040417L;

**public** **void** **doPost**(HttpServletRequest request, HttpServletResponse response) {

**try**

{

System.out.println("get requestion from client");

response.setContentType("text/event-stream"); //SSE header

PrintWriter pw = response.getWriter();

// do something

Thread.sleep(10 \* 1000); // time consume operation

pw.write("data:{code:0, \"message\":\"success\"}\n\n"); // SSE protocol, data:

System.out.println("response sent");

pw.close();

} **catch**(Exception e) {

e.printStackTrace();

}

}

**public** **void** **doGet**(HttpServletRequest request, HttpServletResponse response) {

doPost(request, response);

}

}

HTML代码：  
注意接到数据后关闭客户端，不然浏览器会一直发请求。

<!DOCTYPE unspecified PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<**head**>

<**script** type="text/javascript">

**if** (!!window.EventSource) {

console.log("Event source available");

**var** source = **new** EventSource('/ssedemo');

source.addEventListener('message', **function**(e) {

console.log(e.data);

**if** (document.getElementById('result').innerText) {

document.getElementById('result').innerText = e.data

} **else** {

document.getElementById('result').textContent = e.data

}

source.close(); // close connection, otherwisse browser will request the server constantly

});

source.addEventListener('open', **function**(e) {

console.log("Connection was opened.");

}, false);

source.addEventListener('error', **function**(e) {

**if** (e.readyState == EventSource.CLOSED) {

console.log("Connection was closed.");

} **else** {

console.log(e.readyState);

}

}, false);

} **else** {

console.log("No SSE available");

}

</**script**>

</**head**>

<**body**>

<**div**>Execute a time consume operation....</**div**>

<**div** id="result"></**div**>

</**body**>