



Comet을 이용한 확장

인프라솔루션팀 조영일

Comet이란?

- Another web application model
- Long-held HTTP request
- Push downside from web server
- 반면에, 전통적인 web server는 어떤 메커니즘인가?

Comet의 또 다른 이름

- Ajax Push
- Reverse Push
- Two-way-web
- HTTP Streaming
- HTTP server push

작동 원리

- 요구사항
 - Longer-lived connections
 - 연결 당 필요한 서버쪽 자원의 크기를 최소화
- Comet 서버의 요건
 - asynchronous, event-driven server
 - No select or poll
 - Libevent, epoll, kqueue
 - Many threads per connection

분류

- Off-board
 - PHP로 만드는 경우(threading model이 적합하지 않음)
 - 별도의 threading을 지원하는 언어로 comet 서버를 별도 구축하는 방식
- On-board
 - Erlang, C, Java로 만드는 경우
 - Web server == comet server인 경우

구현 방법

- Polling
- Long Polling
- Forever Frame
- XHR Streaming

구현 방법 – Polling

```
setTimeout(function(){xhrRequest({"foo":"bar"})}, 2000);

function xhrRequest(data){
    var xhr = new XMLHttpRequest();
    // handle the data to send it as parameters on the request
    xhr.open("get", "http://localhost/foo.php", true);
    xhr.onreadystatechange = function(){
        if(xhr.readyState == 4){
            // handle update from server
        }
    };
    xhr.send(null);
}
```

구현 방법 – Long Polling

- 데이터가 변경되면 서버가 보내 줌

```
function longPoll(url, callback) {
  var xhr = new XMLHttpRequest();
  xhr.onreadystatechange = function() {
    if (xhr.readyState == 4) {
      // send another request to reconnect to the server
      callback(xhr.responseText);
      xhr.open('GET', url, true);
      xhr.send(null);
    }
  }
  // connect to the server to open a request
  xhr.open('POST', url, true);
  xhr.send(null);
}
```

구현 방법

- 문제점
 - 연결이 길어진다면, Polling이나 Long Polling이 5만 Q PS의 성능에 도달할 수 있나?

구현 방법 – Forever Frame

```
function foreverFrame(url, callback) {  
  var iframe = body.appendChild(document.createElement("iframe"));  
  iframe.style.display = "none";  
  iframe.src = url + "?callback=parent.foreverFrame.callback";  
  this.callback = callback;  
}
```

And a series of messages from the server might look like this:

```
<script>  
parent.foreverFrame.callback("the first message");  
</script>  
<script>  
parent.foreverFrame.callback("the second message");  
</script>
```

- Hidden iframe 내에서 부모의 콜백을 실행
- 문제점: IE에서 계속 클릭 소리가 남

구현 방법 – Forever Frame

```
function foreverFrame(url, callback){  
  // http://cometdaily.com/2007/11/18/ie-activexhtmlfile-transport-part-ii/  
  // note, do not use 'var tunnel...'  
  htmlfile = new ActiveXObject("htmlfile");  
  htmlfile.open();  
  htmlfile.write(  
    "<html><script>" +  
    "document.domain=" + document.domain + ";" +  
    "</script></html>");  
  htmlfile.close();  
  var ifrDiv = tunnel.createElement("div");  
  htmlfile.body.appendChild(ifrDiv);  
  ifrDiv.innerHTML = "<iframe src=" + url + "></iframe>";  
  foreverFrame.callback = callback;  
}
```

- ActiveXObject 사용해서 클릭 소리를 없앴

구현 방법 – XHR Streaming

```
function xhrStreaming(url, callback){
  xhr = new XMLHttpRequest();
  xhr.open('POST', url, true);
  var lastSize;
  xhr.onreadystatechange = function(){
    var newTextReceived;
    if(xhr.readyState > 2){
      // get the newest text
      newTextReceived =
        xhr.responseText.substring(lastSize);
      lastSize = xhr.responseText.length;
      callback(newTextReceived);
    }
    if(xhr.readyState == 4){
      // create a new request if the response is finished
      xhrStreaming(url, callback);
    }
  }
  xhr.send(null);
}
```

구현 방법 – XHR Streaming

- 제한점
 - 서버: 스트리밍 로직, 버퍼 관리 등의 복잡한 문제점을 알아서 해결해야 함
 - 클라이언트: 메모리 부족

대안

- WebSocket
 - HTML5에서 지원
 - 브라우저에서 사용할 수 있는 TCP 소켓
 - 지속적인 연결을 맺고 onmessage 콜백을 통해 메시지를 받을 수 있음
- Bayeux protocol
- BOSH protocol
- JSONRequest object

크로스 도메인

- 브라우저마다 조금씩 상황이 다를 수 있음
- Forever Frame은 이슈 없음
- HTML5 postMessage 기술을 활용할 것
- XHR을 쓴다면 JSONP를 고려해 볼 것



T H A N K Y O U