



25

Ajax 연습 3 – Reverse Ajax

모던 웹을 위한 Javascript jQuery 입문

25.1 Reverse Ajax 개요

❖ Reverse Ajax

- 서버에서 원할 때 클라이언트에게 데이터를 제공하는 방법
- 대부분의 SNS에서 사용
 - Facebook 이 대표적인 예

그림 25-1 글 입력과 글 알림



❖ Reverse Ajax 기법

- Polling 기법
- Long Polling 기법
- Streaming(Comet) 기법



25.1 Reverse Ajax 개요

❖ Streaming 기법

- 가장 완벽한 방식의 Reverse Ajax
- 정말 서버가 원하는 순간에 데이터를 클라이언트에게 전송

그림 25-2 Streaming(Comet) 기법

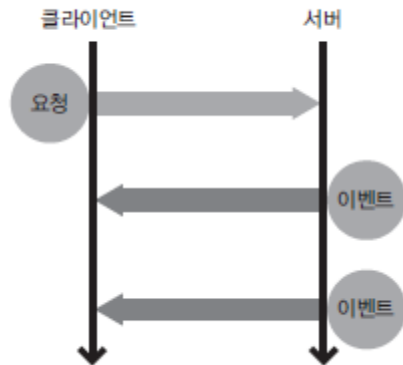
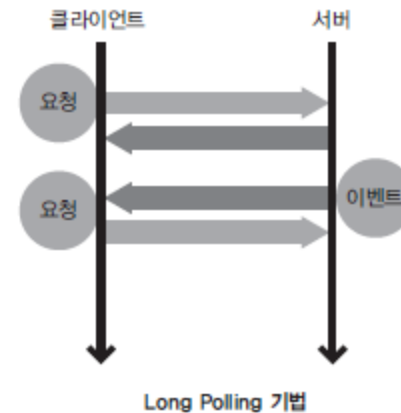
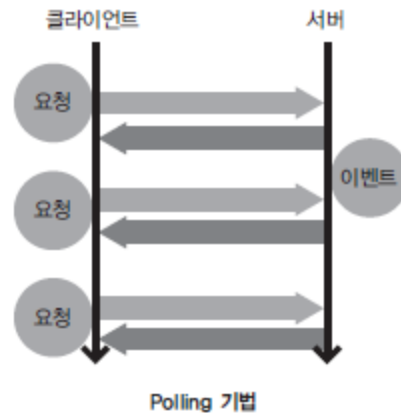


그림 25-3 Pooling 기법과 Long Polling 기법

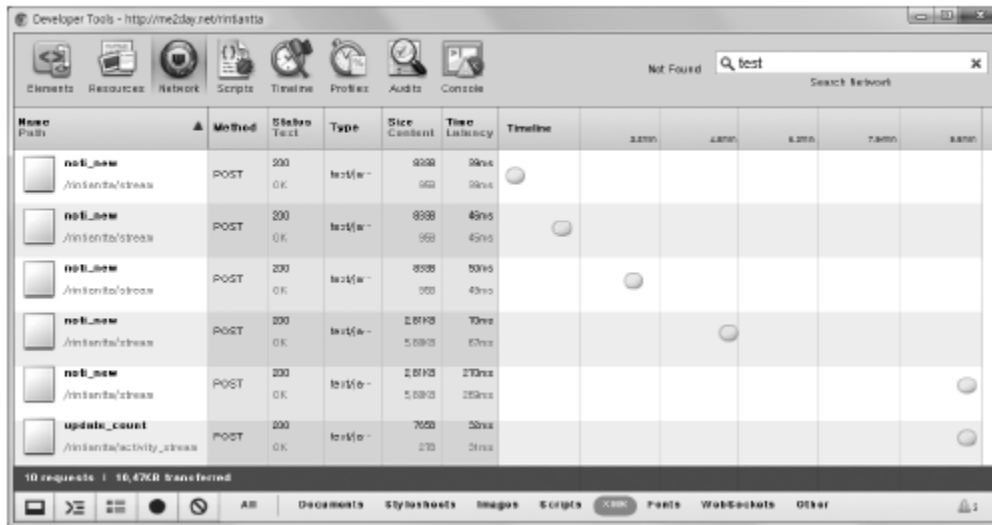


25.1 Reverse Ajax 개요

❖ Polling 기법

- 주기적으로 타이머를 사용해 서버에게 변경된 사항이 있는지 계속 Ajax 요청 사용해 물어보는 방식
- Ex) 네이버 미투데이의 네트워크 요청

그림 25-4 일정 시간마다 Ajax를 요청



25.1 Reverse Ajax 개요

❖ Long Polling 기법

- 클라이언트가 서버에게 Ajax 요청
- 서버가 이벤트가 발생하기 전까지 붙잡고 있는 방식
- 서버에 부하가 걸리는 단점

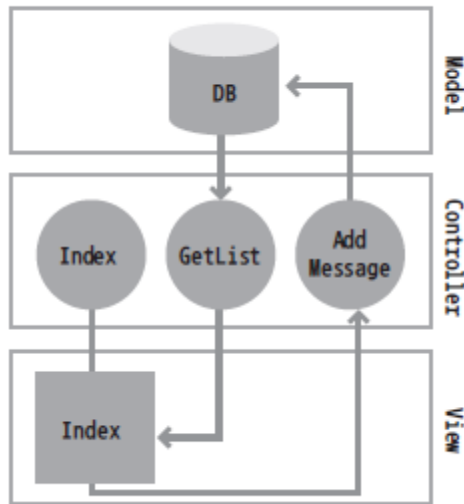


25.1 Reverse Ajax 개요

❖ 채팅 애플리케이션 프로젝트

- 액션 GetList - 데이터베이스에서 채팅 목록 가져옴
- 액션 AddMessage는 데이터베이스에 메시지 추가

그림 25-5 프로젝트의 구성



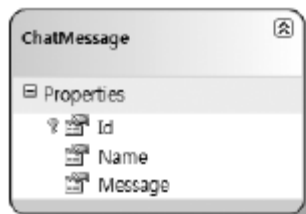
25.2 데이터베이스 만들기

❖ 프로젝트 AjaxChat 생성

그림 25-6 데이터베이스 파일 생성



그림 25-7 ChatMessage 테이블



25.3 웹 서비스 만들기

❖ 액션 AddMessage

- 매개 변수 name과 message 입력 받고 데이터베이스에 추가

❖ 액션 GetList

- 데이터베이스의 내용 JSON으로 제공

코드 25-2 HomeController.cs: 액션 AddMessage 구성

```
public ActionResult AddMessage(string name, string message)
{
    // 추가할 데이터를 생성합니다.
    var willInsert = new ChatMessage();
    willInsert.Name = name;
    willInsert.Message = message;

    // 데이터베이스에 데이터를 추가합니다.
    var db = new DataClassesDB();
    db.ChatMessage.InsertOnSubmit(willInsert);
    db.SubmitChanges();

    // 리턴합니다.
    return Content("SUCCESS");
}
```

코드 25-3 HomeController.cs: 액션 AddMessage 구성

```
public ActionResult GetList()
{
    var db = new DataClassesDB();
    return Json(db.ChatMessage, JsonRequestBehavior.AllowGet);
}
```



❖ 뷰 페이지

- 컨트롤러 HomeController에서 액션 Index를 선택하고 뷰 생성
- 생성된 뷰의 body 태그에 입력 양식과 채팅 메시지 출력할 div 태그 생성

코드 25-4 Index.cshtml: body 태그 구성

```
<body>
  <form id="write_form">
    <table>
      <tr>
        <td>닉네임</td>
        <td><input type="text" name="name"/></td>
      </tr>
      <tr>
        <td>메시지</td>
        <td><textarea name="message"/></td>
      </tr>
    </table>
    <input type="submit" value="글쓰기"/>
  </form>
  <hr/>
  <div id="output">

</div>
</body>
```



❖ script 태그

- 입력 양식에 submit 이벤트 연결하고 기본 이벤트 제거
- post() 메서드 사용해 데이터 전송

코드 25-5 Index.cshtml: submit 이벤트 연결

```
<script>
    $(document).ready(function () {
        // 이벤트를 연결합니다.
        $('#write_form').submit(function (event) {
            // Ajax를 수행합니다.
            $.post('/Home/AddMessage', $(this).serialize());

            // 디폴트 이벤트를 제거합니다.
            event.preventDefault();
        });

        // 1초마다 함수를 실행합니다.
        setInterval(function () {

        }, 1000);
    });
</script>
```



❖ Polling 기법

- setInterval() 함수 사용해 1초마다 변경 사항 검사
 - ajax 사용해 데이터 div 태그에 추가
 - 다른 요청임을 구분하려고 just 매개 변수에 지속적으로 다른 값 넣어 요청

코드 25-6 Index.cshtml: setInterval() 함수 사용

```
setInterval(function () {  
    $.getJSON('/Home/GetList', { just: new Date().getTime() }, function (data) {  
        // #output의 내용물을 제거합니다.  
        $('#output').empty();  
  
        // #output에 내용물을 추가합니다.  
        $.each(data, function (index, item) {  
            var output = '';  
            output += '<h2>' + data[i].Name + '</h2>';  
            output += '<p>' + data[i].Message + '</p>';  
  
            $('<div></div>').html(output).prependTo('#output');  
        });  
    });  
}, 1000);
```



25.4 Ajax 사용

❖ 실행 결과

그림 25-8 Polling을 사용한 채팅





Thank You !

모던 웹을 위한 Javascript jQuery 입문