24 Ajax 연습 2 – 크로스 도메인

모던 웹을 위한 Javascript jQuery 입문

한빛미디어 Hanbit Media, Inc.

0. 서론

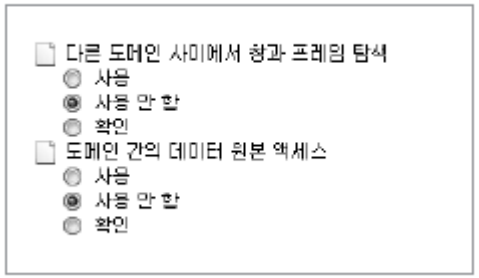
❖ 다른 서버의 데이터 가져오기

- XMLHttpRequest 객체
  - 보안상 제한으로 인해 자바스크립트 파일을 가져왔던 서버하고만 통신
  - 다른 서버와 통신하는 경우?
    - **브라우저에서 입력한 정보를 사용자가 모르는 사이에 다른 서버로 전송하는 것 가능**

❖ 크로스 도메인 통신

- 불특정 다수의 도메인 서버에 접근하는 것
- 인터넷 익스플로러
  - [도구]→[인터넷 옵션]→[보안 설정]→[사용자 지정 수준]을 선택

그림 24-1 인터넷 익스플로러의 보안 설정



24.1 크로스 도메인 개요

❖ JSONP 방식

- JSON 파일에만 적용할 수 있는 방법
- 클라이언트에서 다른 도메인에 마음대로 접근할 수 없음
- 서버 측에서 접근하는 것은 문제가 없으므로 사용할 수 있는 방법

그림 24-2 외부 도메인에서 데이터 가져오기

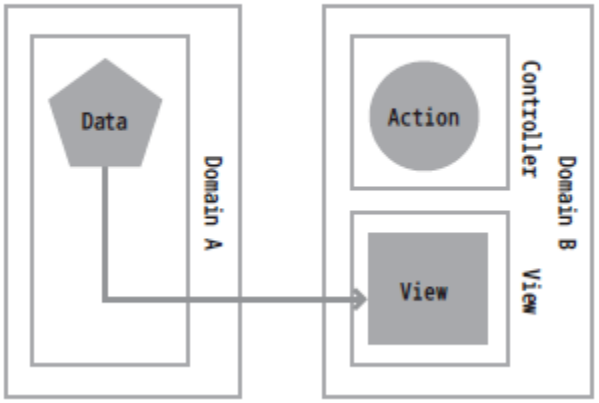
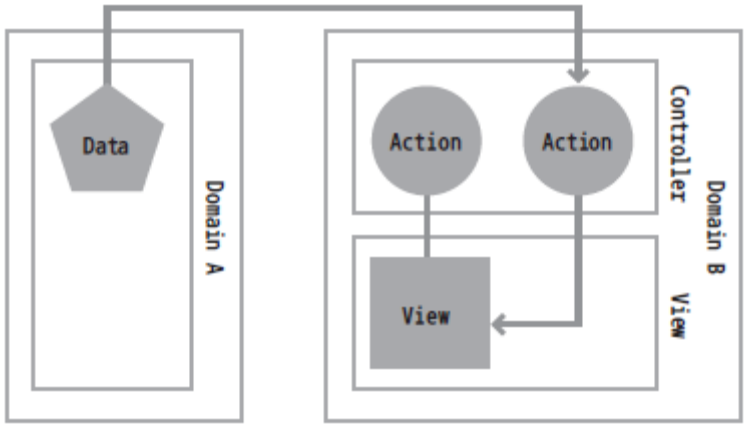


그림 24-3 보완된 형태의 크로스 도메인 통신



24.2 JSONP 방식

❖ JSONP (JSON with Padding)

- 크로스 도메인의 제약에서 벗어나 동적으로 데이터를 서버에 전달하고 응답 받아 처리하려고 고안한 방법
- <script src="http://code.jquery.com/jquery-1.7.js"></script>
  - 자바 스크립트는 외부 도메인 사용 가능
- 네이버 미투데이 서비스에서 제공하는 사용자의 포스트 가져오기
  - http://me2day.net/api/get_posts/[미투데이 사용자아이디].[응답형식]
  - http://dev.naver.com/openapi/apis/me2day/me2api_intro 의 미투데이 API 문서 참조

코드 24-1 JSONP 기본 개념

```
<head>
    <script src="http://code.jquery.com/jquery-1.7.js"></script>
    <script src="http://me2day.net/api/get_posts/rintiantta.json"></script>
</head>
```

24.2 JSONP 방식

❖ JSONP 사용

코드 24-2 JSONP의 사용

```
<head>
    <script src="http://code.jquery.com/jquery-1.7.js"></script>
    <script>
        function jsonpTest(data) {
            alert(data);
        }
    </script>
    <script
src="http://me2day.net/api/get_posts/rintiantta.json?callback=jsonpTest"></script>
</head>
```

그림 24-4 실행 결과

24.2 JSONP 방식

❖ jQuery를 통한 간단한 JSONP 사용

코드 24-3 jQuery를 사용한 JSONP

```
<script>
    $(document).ready(function () {
        $.Ajax('http://me2day.net/api/get_posts/rintiantta.json', {
            dataType: 'jsonp',
            success: function (data) {
                alert(data);
            }
        });
    });
</script>
```

24.2 JSONP 방식

❖ 데이터로 문서객체 생성 후 Body 태그에 출력

코드 24-4 크로스 도메인 Ajax 통신을 사용한 문서 객체 추가

```
<script>
    $(document).ready(function () {
        $.Ajax('http://me2day.net/api/get_posts/rintiantta.json', {
            dataType: 'jsonp',
            success: function (data) {
                $.each(data, function (index, item) {
                    // 문서 객체를 생성합니다.
                    var a = $('<h2></h2>').html(item.author.id);
                    var b = $('<img />').attr('src', item.author.face);
                    var c = $('<img />').attr('src', item.icon);
                    var d = $('<p></p>').addClass('date').html(item.pubDate);
                    var e = $('<p></p>').html(item.body);
                    var f = $('<img />').attr('src', item.media.photoUrl);

                    // body 태그에 내용을 추가합니다.
                    $('<div></div>').append(a, b, c, d, e, f).appendTo('body');
                });
            }
        });
    });
</script>
```

❖ style 태그 구성

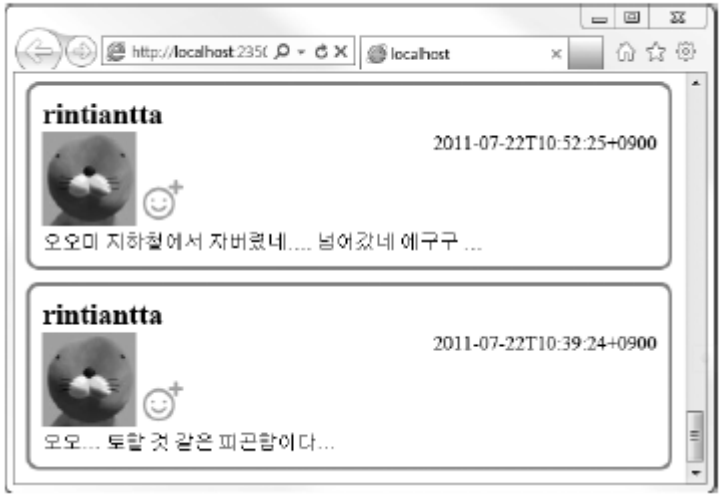
코드 24-5 style 태그 구성

```
<style>
    * { margin:0px; padding:0px; }
    div  {
        margin:10px;
        padding:10px;

        border-width:3px;
        border-style:solid;
        border-color:#9F9F9F;
        border-radius:10px;
    }

    div >.date {
        float:right;
    }
</style>
```

24.2 JSONP 방식

❖ 예제 실행 화면

그림 24-5 실행 결과



getJSON() 메서드를 사용하고 싶을 때는 다음과 같이 입력합니다. 자동으로 jQuery가 모든 것을 수행해준답니다.

```
$.getJSON('http://me2day.net/api/get_posts/rintiantta.json?callback=?',
    function (data) {

});
```

24.3 RSS 웹 서비스 만들기

❖ XML 문서나 HTML 페이지를 전송 받으려면?

그림 24-6 보완된 형태의 크로스 도메인 데이터 가져오기

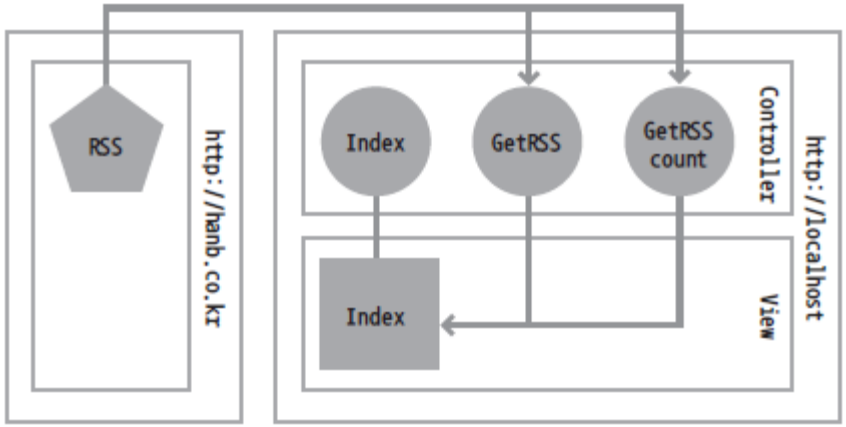
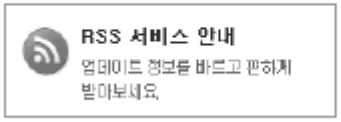


그림 24-7 한빛미디어의 RSS 서비스



❖ ASP.NET MVC 3프로젝트 생성

- 액션 GetRSS 생성
  - 한빛미디어의 RSS 문서를 가져옴

코드 24-7 HomeController.cs: 액션 GetRSS 생성

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        return View();
    }

    public ActionResult GetRSS()
    {

    }
}
```

24.3 RSS 웹 서비스 만들기

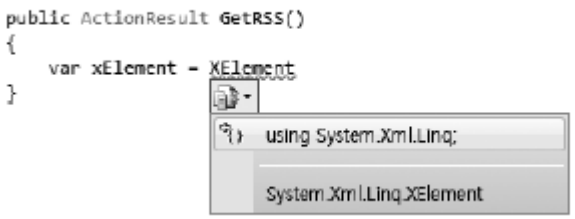
❖ 액션 GetRSS 생성

- XElement 클래스 부분에 에러 발생 - using 구문 추가

코드 24-8 HomeController.cs: 다른 도메인의 XML 가져오기

```
public ActionResult GetRSS()
{
    var xElement = XElement.Load("http://www.hanb.co.kr/sync/rss_newbook.xml");
    return Content(xElement.ToString());
}
```

그림 24-8 using System.Xml.Linq



24.3 RSS 웹 서비스 만들기

❖ 실행 결과

- 데이터 크기가 크므로 XML 대신 JSON으로 변경

그림 24-9 가져온 RSS 데이터

24.3 RSS 웹 서비스 만들기

❖ 예제 코드 24-9]

- 페이징 기능과 JSON을 리턴 하는 기능을 추가한 액션 GetRSS
- 원하는 데이터를 선별하는 부분 존재

코드 24-9 HomeController.cs: XML 서비스 구성

```
public ActionResult GetRSS(int start, int count)
{
    // XML을 가져옵니다.
    var xElement = XElement.Load("http://www.hanb.co.kr/sync/rss_newbook.xml");

    // 원하는 데이터를 선별합니다.
    var data = from item in xElement.Descendants("item")
               select new
               {
                   title = item.Element("title").Value,
                   link = item.Element("link").Value,
                   description = item.Element("description").Value,
                   author = item.Element("author").Value,
                   pubDate = item.Element("pubDate").Value
               };

    // 데이터를 JSON으로 만듭니다.
    return Json(data.Skip(start * count + 1).Take(count), JsonRequestBehavior.
AllowGet);
}
```

24.3 RSS 웹 서비스 만들기

❖ 예제 코드 24-10]

- 액션 GetRSSCount
  - RSS 문서의 item 요소 개수 리턴
  - 한빛미디어의 RSS 문서는 기본적으로 25개의 item 요소 가짐

코드 24-10 HomeController.cs: 액션 GetRSSCount 생성

```
public ActionResult GetRSSCount()
{
    // XML을 가져옵니다.
    var xElement = XElement.Load("http://www.hanb.co.kr/sync/rss_newbook.xml");
    var data = from item in xElement.Descendants("item") select item;

    // 데이터를 JSON으로 만듭니다.
    return Json(data.Count(), JsonRequestBehavior.AllowGet);
}
```

24.4 Ajax 사용

❖ 페이징과 출력

- 액션 GetRSSCount
  - RSS 문서의 items 요소의 개수 가져와 페이지 번호 생성
- 액션 GetRSS
  - JSON 데이터 가져와 화면에 출력

코드 24-12 Index.cshtml : Ajax 사용

```
<script>
    $(document).ready(function () {
        // 페이지당 내용물의 수를 정의합니다.
        var count = 5;

        // 페이지를 나눕니다.
        $.get('/Home/GetRSSCount', function (data) {
            // 숫자로 바꿉니다.
            var perPage = Math.ceil(Number(data) / count);

            // 페이지를 만듭니다.
            for (var i = 0; i < perPage; i++) {
                $('<h1></h1>').html(i).appendTo('#page_selector');
            }
        });

        // 이벤트를 연결합니다.
        $('#page_selector > h1').live('click', function () {
            // 클릭한 페이지를 가져옵니다.
            var start = $(this).html();
```

❖ 페이징과 출력 (2)

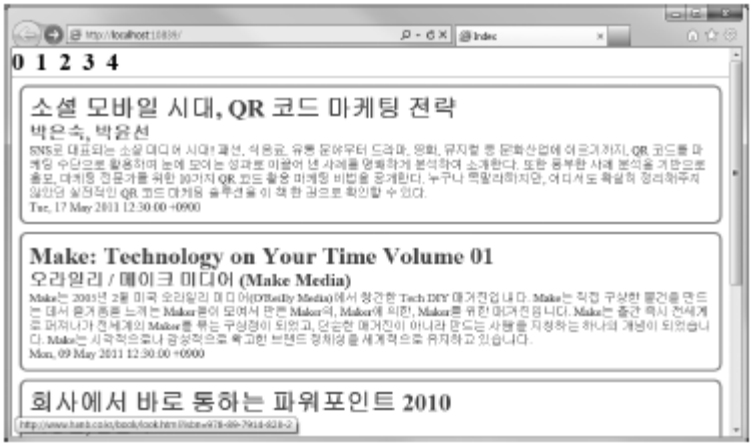
```
        // 데이터를 추가합니다.
        $.getJSON('/Home/GetRSS', {
            start: start,
            count: count
        }, function (data) {
            // #output의 내용을 지웁니다.
            $('#output').empty();

            // #output에 내용을 추가합니다.
            $.each(data, function (index, item) {
                var output = '';
                output += '<a href="' + item.link + '">';
                output += '<div link>';
                output += '    <h1>' + item.title + '</h1>';
                output += '    <h2>' + item.author + '</h2>';
                output += '    <p>' + item.description + '</p>';
                output += '    <p>' + item.pubDate + '</p>';
                output += '</div>';
                output += '</a>';

                $('#output').append(output);
            });
        });
    });
});
</script>
```

❖ style 태그 구성 – 출력

코드 24-13 Index.cshtml: style 태그 구성

```
<style>
    * { margin:0px; padding:0px; }
    #page_selector { overflow:hidden; }
    #page_selector > h1 {
        padding-right:15px;
        float:left;
    }
    #output > a { text-decoration:none; }
    #outputdiv  {
        margin:10px;
        padding:10px;

        border-width:3px;
        border-style:solid;

        border-color:#9F9F9F;
        border-radius:10px;
    }
</style>
```

그림 24-10 실행 결과



Thank You !

모던 웹을 위한 Javascript jQuery 입문

한빛미디어 Hanbit Media, Inc.