



# 23 Ajax 연습 1 - 데이터베이스

모던 웹을 위한 Javascript jQuery 입문

# 23.1 샘플 데이터베이스 다운로드

## ❖ 마이크로소프트가 제공하는 샘플 데이터베이스 사용

- <http://msftdbprodsamples.codeplex.com/releases/view/37109>
- Other Available Download의
  - 'SQL2008.AdventureWorksLT2008 Only Database.zip' 사용  
– **SalesLT.Product 테이블**

그림 23-1 샘플 데이터베이스 파일 받기

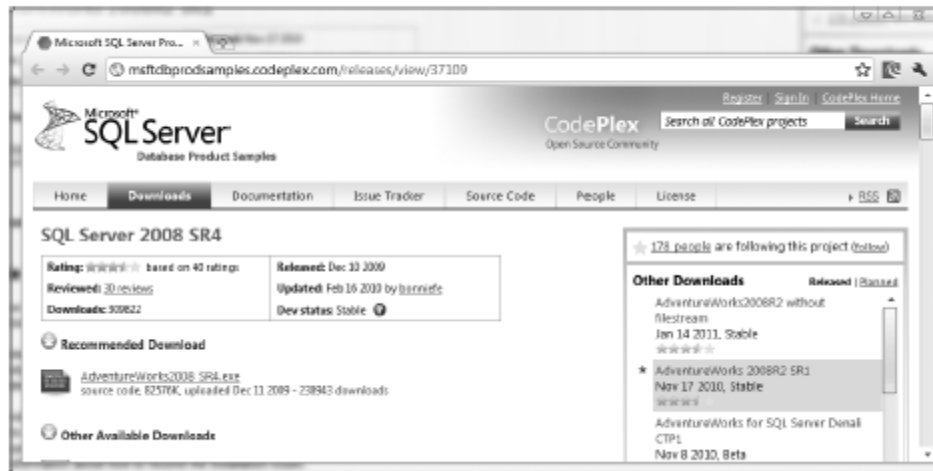
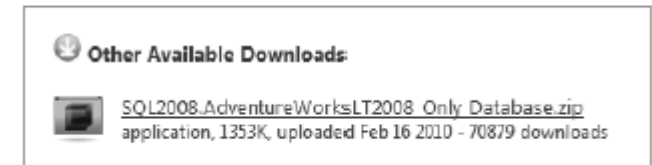


그림 23-2 Other Available Downloads



## ❖ 프로젝트 구성

그림 23-3 App\_Data 폴더 생성

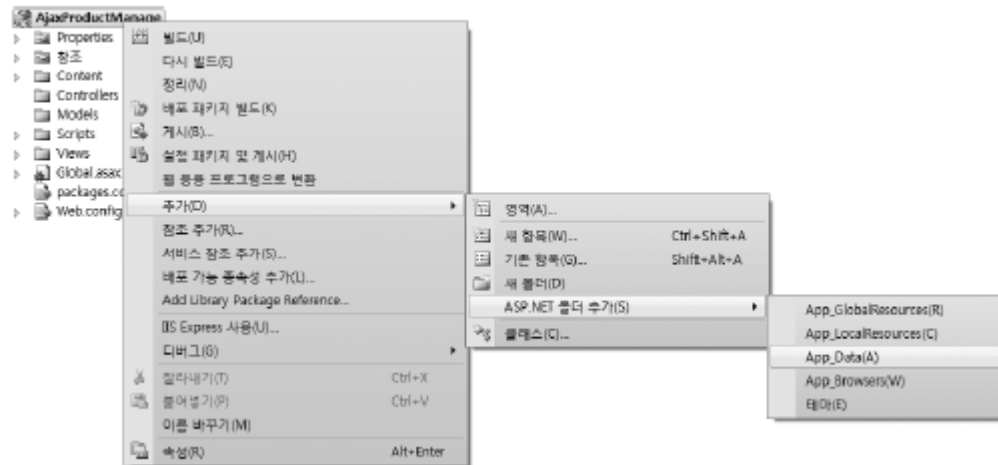
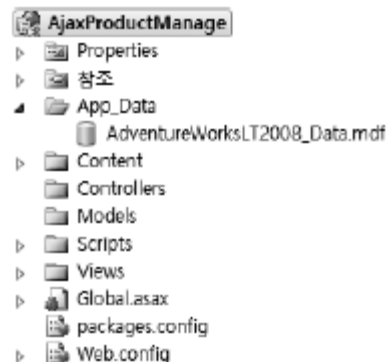


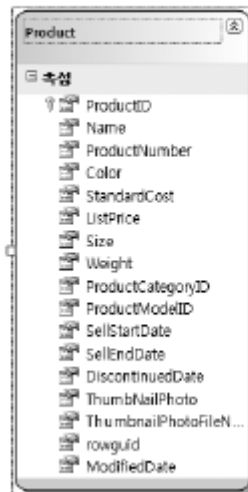
그림 23-4 AdventureWorksLT2008\_Data.mdf 파일 추가



## ❖ 데이터베이스 서비스 만들기

- AdventureWorks라는 이름으로 LINQ To SQL 파일 생성
- 그림 23-5처럼 테이블 구성
  - AdventureWorks.dbml의 Name 속성
    - 사용하기 쉽게 AdventureWorksDB로 지정

그림 23-5 AdventureWorks.dbml 구성



## ❖ 데이터베이스 서비스 만들기 (2)

- 기본 LINQ To SQL 구성 완료
- 데이터를 제공하는 액션 GetProductList 와 각 제품의 이미지를 제공하는 액션 GetProductImage 생성

코드 23-1 HomeController.cs: 액션 GetProductList와 액션 GetProductImage 생성

코드 23-2 HomeController.cs: 액션 구성

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

    public ActionResult GetProductList()
    {
        var adventureWorksDB = new AdventureWorksDB();
        return Json(adventureWorksDB.Product, JsonRequestBehavior.AllowGet);
    }

    public FileContentResult GetProductImage(int id)
    {
        var adventureWorksDB = new AdventureWorksDB();
        var willReturn = adventureWorksDB.Product.Single(x => x.ProductID == id);
        return File(willReturn.ThumbnailPhoto.ToArray(), "image/gif");
    }
}
```



## ❖ 데이터베이스 서비스 만들기 (3)

- 프로젝트 실행
- <http://localhost:PORT/Home/GetProductImage/710> 접속
  - 그림을 가져오는 것이 실패한 것이 아님
  - AdventureWorks 데이터베이스에 들어 있는 그림이 'No Image Available'

그림 23-6 액션 GetProductImage의 710번 접속 결과



## ❖ index.cshtml 작성

- HTML 페이지 구성
- div 태그 위에 데이터가 올라감

코드 23-3 Index.cshtml: HTML 페이지 구성

```
@{ Layout = null; }  
<!DOCTYPE html>  
<html>  
<head>  
    <title>Index</title>  
    <style>  
  
    </style>  
    <script src="http://code.jquery.com/jquery-1.7.js"></script>  
    <script>  
        $(document).ready(function () {  
  
            });  
    </script>  
</head>  
<body>  
    <div id="wrap">  
  
    </div>  
</body>  
</html>
```



### ❖ 액션 GetProductList

- 데이터 요청하고 each() 메서드 사용해 각 데이터 추출
- 문서 객체 만들어 이전에 만든 div 태그에 추가

코드 23-4 Index.cshtml: script 태그 구성

```
<script>
    $(document).ready(function () {
        $.getJSON('/Home/GetProductList', function (data) {
            $(data).each(function (index, item) {
                // 변수를 선언합니다.
                var imageURL = '/Home/GetProductImage/' + item.ProductID;
                var output = '';
                output += '<div class="item">';
                output += '    <h3>' + item.ProductID + ' : ' + item.Name + '</h3>';
                output += '    ';
                output += '    <p>' + 'productNumber: ' + item.ProductNumber + '</p>';
                output += '    <p>' + 'color: ' + item.Color + '</p>';
                output += '    <p>' + 'standardCost: ' + item.StandardCost + '</p>';
                output += '</div>';

                // 문서 객체에 추가합니다.
                $('#wrap').append(output);
            });
        });
    });
</script>
```





## ❖ Style 태그 구성 & 전체 실행 결과

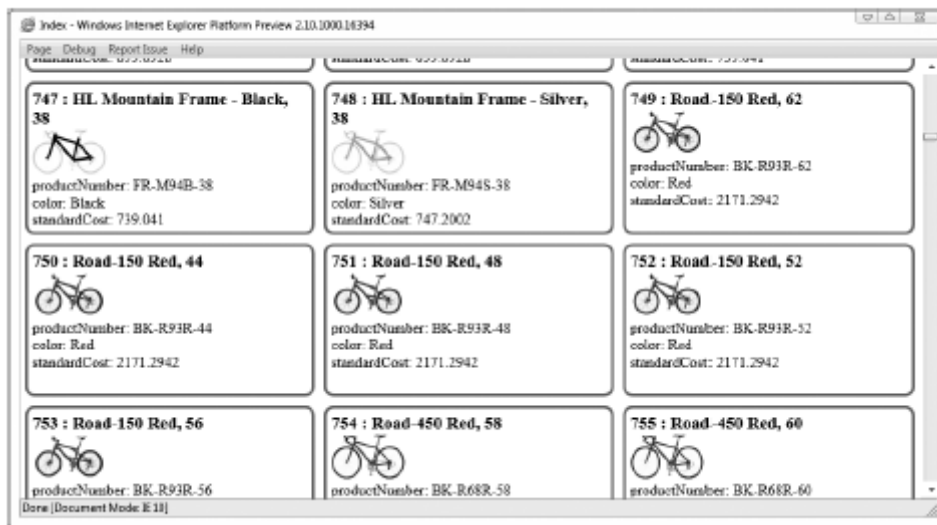
코드 23-5 Index.cshtml: style 태그 구성

```
<style>
* { margin:0px; padding:0px; }
#wrap { overflow:hidden; }

.item
{
    margin:5px; padding:5px;
    width:300px; height:150px;
    float:left;

    border-width:3px;
    border-style:solid;
    border-color:Gray;
    border-radius:10px;
}
</style>
```

그림 23-7 실행 결과



## ❖ 페이징이란?

- 한 페이지에 정해진 개수의 데이터만을 보여주는 것
- 데이터가 몇 개 있는지 알려주는 웹 서비스와 정해진 범위의 데이터 가져오는 웹 서비스 필요

코드 23-7 HomeController.cs: 액션 GetTotalProductCount 구성

```
public ActionResult GetTotalProductCount()
{
    var adventureWorksDB = new AdventureWorksDB();
    return Content(adventureWorksDB.Product.Count().ToString());
}
```

코드 23-8 HomeController.cs: 액션 GetPagingProductList 구성

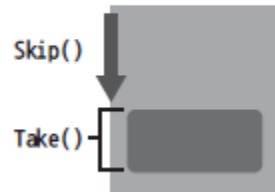
```
public ActionResult GetPagingProductList(int nowPage, int perPage)
{
    var adventureWorksDB = new AdventureWorksDB();
    var data = adventureWorksDB.Product.Skip(nowPage * perPage).Take(perPage);
    return Json(data, JsonRequestBehavior.AllowGet);
}
```



## ❖ Skip() 메서드/Take() 메서드

- 특정 부분의 데이터 가져올 수 있는 메서드

그림 23-8 Skip() 메서드와 Take() 메서드



## ❖ Index.cshtml

- body 태그에 페이지 숫자 표시할 수 있는 ul 태그 두 개 구성
- 리스트는 가로로 구성

코드 23-9 Index.cshtml: body 태그 구성

```
<body>
  <ul class="paging_button"></ul>
  <div id="wrap">
    <div>
      <ul class="paging_button"></ul>
    </div>
  </div>
```

코드 23-10 Index.cshtml: style 태그에 추가할 내용

```
ul { overflow:hidden; }
li { float:left; list-style:none; padding:10px; cursor:pointer; }
```



## ❖ Script 태그 구성

### ■ getItems() 함수

- 매개 변수로 nowPage와 itemsPerPage 입력
- 액션 GetPagingProductList와 대응하는 매개 변수
  - 정해진 양의 데이터를 액션 GetPagingProductList에 요청
  - 데이터 가져와 화면에 출력

코드 23-11 Index.cshtml : script 태그 구성

```
<script>
$(document).ready(function () {
    // 내부 함수 선언
    function getItems(nowPage, itemsPerPage) {

    }

    // 페이지와 관련된 정보를 가지는 변수
    // 페이지 버튼 추가
    // 페이지 버튼 이벤트 연결
    // 0으로 초기 페이지 설정
});
</script>
```



## ❖ Script 태그 구성

### ■ 내부 함수 선언

코드 23-12 Index.cshtml: 내부 함수 선언

```
// 내부 함수 선언
function getItems(nowPage, itemsPerPage) {
    $.getJSON('/Home/GetPagingProductList', {
        nowPage: nowPage,
        perPage: itemsPerPage
    }, function (data) {
        $(data).each(function (index, item) {
            // 변수를 선언합니다.
            var imageURL = '/Home/GetProductImage/' + item.ProductID;
            var output = '';
            output += '<div class="item">';
            output += '    <h3>' + item.ProductID + ' : ' + item.Name + '</h3>';
            output += '    ';
            output += '    <p>' + 'productNumber: ' + item.ProductNumber + '</p>';
            output += '    <p>' + 'color: ' + item.Color + '</p>';
            output += '    <p>' + 'standardCost: ' + item.StandardCost + '</p>';
            output += '</div>';

            // 문서 객체에 추가합니다.
            $('#wrap').append(output);

        });
    });
}
```



## ❖ Script 태그 구성

### ■ 페이지 버튼 추가

- 페이지 개수는 정수 - Math.ceil() 메서드 사용
- 각 페이지 버튼에 data-page 속성 입력

코드 23-13 Index.cshtml: 페이지 버튼 추가

```
// 페이지와 관련된 정보를 가지는 변수
var itemsPerPage = 12;

// 페이지 버튼 추가
$.get('/Home/GetTotalProductCount', function (data) {
    var totalItem = Number(data);
    var totalPages = Math.ceil(totalItem / itemsPerPage);

    for (var i = 0; i < totalPages; i++) {
        var $add = $('<li/>').html(i).attr('data-page', i);
        $('#paging_button').append($add);
    }
});
```



## ❖ 이벤트 연결

- data-page 속성 바탕으로 getItem() 메서드 실행
- jQuery의 Ajax 메서드들은 비동기적으로 데이터 요청
  - 이벤트를 연결한 순간에 문서 객체 생성 여부는 정해지지 않음
    - **live() 메서드로 연결**
- 초기 페이지 0으로 설정

코드 23-14 Index.cshtml: 이벤트 연결

```
// 페이지 버튼 이벤트 연결
$('#paging_button li').live('click', function () {
    // 원래 있던 내용 제거
    $('#wrap').empty();

    // 새로 내용을 넣습니다.
    var nowPage = $(this).attr('data-page');
    getItem(nowPage, itemsPerPage);
});
```

코드 23-15 Index.cshtml: 초기 페이지 설정

```
// 0으로 초기 페이지 설정
getItem(0, itemsPerPage);
```





## ❖ 페이징 최종 예제

코드 23-16 Index.cshtml: 최종 완성

```
<script>
$(document).ready(function () {
    // 내부 함수 선언
    function getItems(nowPage, itemsPerPage) {
        $.getJSON('/Home/GetPagingProductList', {
            nowPage: nowPage,
            perPage: itemsPerPage
        }, function (data) {
            $(data).each(function (index, item) {
                // 변수를 선언합니다.
                var imageURL = '/Home/GetProductImage/' + item.ProductID;
                var output = '';
                output += '<div class="item">';
                output += '    <h3>' + item.ProductID + ' : ' + item.Name + '</h3>';
                output += '    ';
                output += '    <p>' + 'productNumber: ' + item.ProductNumber + '</p>';
                output += '    <p>' + 'color: ' + item.Color + '</p>';
                output += '    <p>' + 'standardCost: ' + item.StandardCost + '</p>';
                output += '</div>';

                // 문서 객체에 추가합니다.
                $('#wrap').append(output);
            });
        });
    }
});
</script>
```



## ❖ 페이징 최종 예제 (2)

// 페이지와 관련된 정보를 가지는 변수

```
var itemsPerPage = 12;
```

// 페이지 버튼 추가

```
$.post('/Home/GetTotalProductCount', function (data) {  
    var totalItem = Number(data);  
    var totalPages = Math.ceil(totalItem / itemsPerPage);  
    for (var i = 0; i < totalPages; i++) {  
        var $add = $('<li/>').html(i).attr('data-page', i);  
        $('#paging_button').append($add);  
    }  
});
```

// 페이지 버튼 이벤트 연결

```
$('#paging_button li').live('click', function () {  
    // 원래 있던 내용 제거  
  
    $('#wrap').empty();  
  
    // 새로 내용을 넣습니다.  
    var nowPage = $(this).attr('data-page');  
    getItems(nowPage, itemsPerPage);  
});
```

// 0으로 초기 페이지 설정

```
getItems(0, itemsPerPage);
```

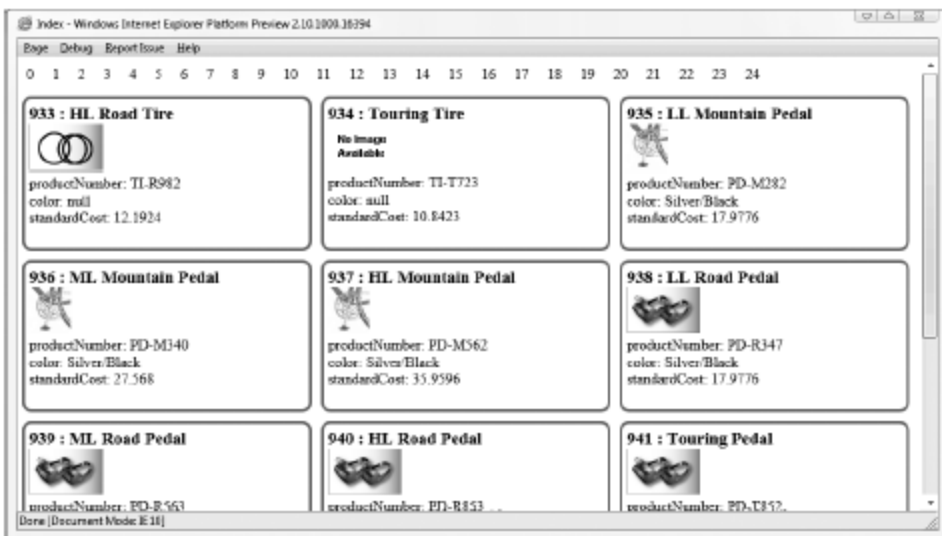
```
});
```

```
</script>
```



## ❖ 프로젝트 실행 화면

그림 23-9 한 페이지에 12개씩 제품이 소개되고 페이징 버튼을 사용해 이동할 수 있습니다.





# Thank You !

모던 웹을 위한 Javascript jQuery 입문