



27

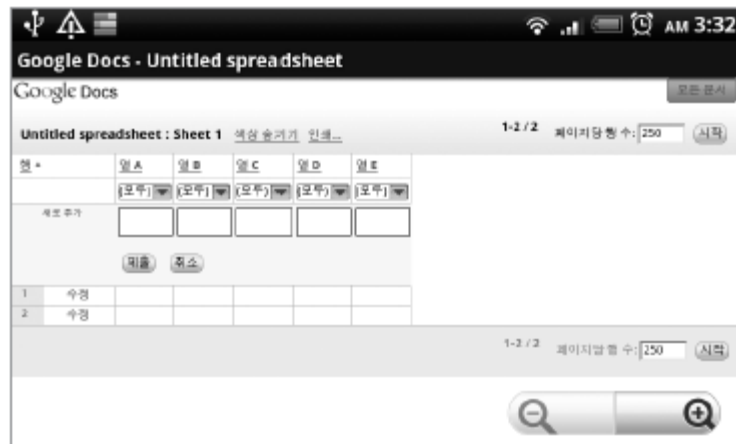
jqGrid 플러그인

모던 웹을 위한 Javascript jQuery 입문

❖ jqGrid 플러그인

- 그리드를 손쉽게 만들어주는 jQuery의 플러그인
- 그림 27-1의 그리드는 Google Docs 웹 애플리케이션
 - <https://docs.google.com/> 접속해 작업 가능

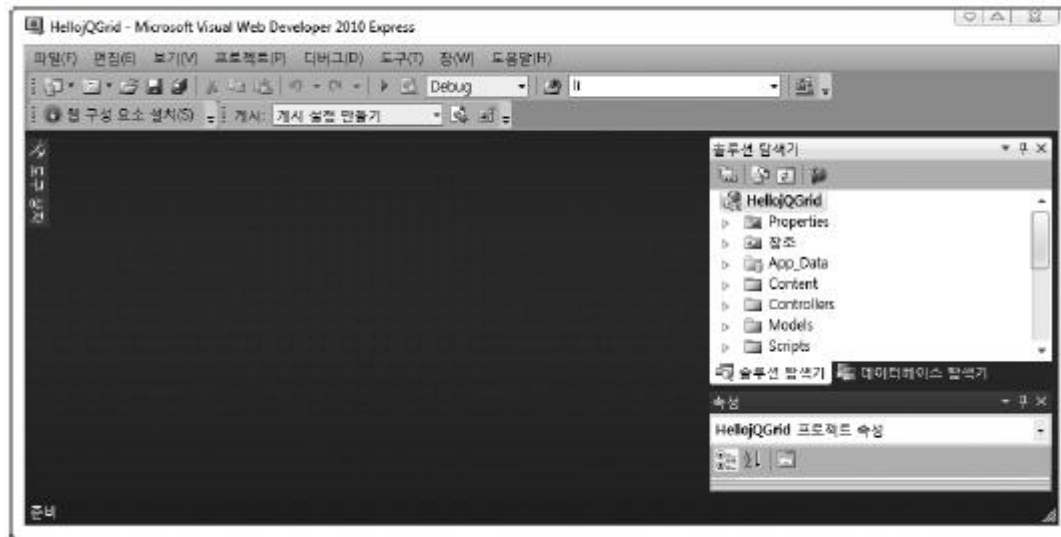
그림 27-1 안드로이드 애플리케이션 Google Docs의 그리드



❖ 도서관리 그리드 작성

- HellojqGrid라는 이름의 프로젝트 생성
- HomeController를 만들고 Index 뷰까지 생성

그림 27-2 프로젝트 생성



❖ Dynamic LINQ To SQL

- Visual Web Developer 2010 Express에 내장돼 있지 않음
- <http://msdn.microsoft.com/en-us/vcsharp/bb894665.aspx> 에서 다운
- 기존의 LINQ To SQL로 구현하기 어려운 기능을 쉽게 만들 수 있음
- LinqSamples/DynamicQuery/DynamicQuery/Dynamic.cs
 - Controllers 폴더에 드래그해 이동

그림 27-3 <http://msdn.microsoft.com/en-us/vcsharp/bb894665.aspx>

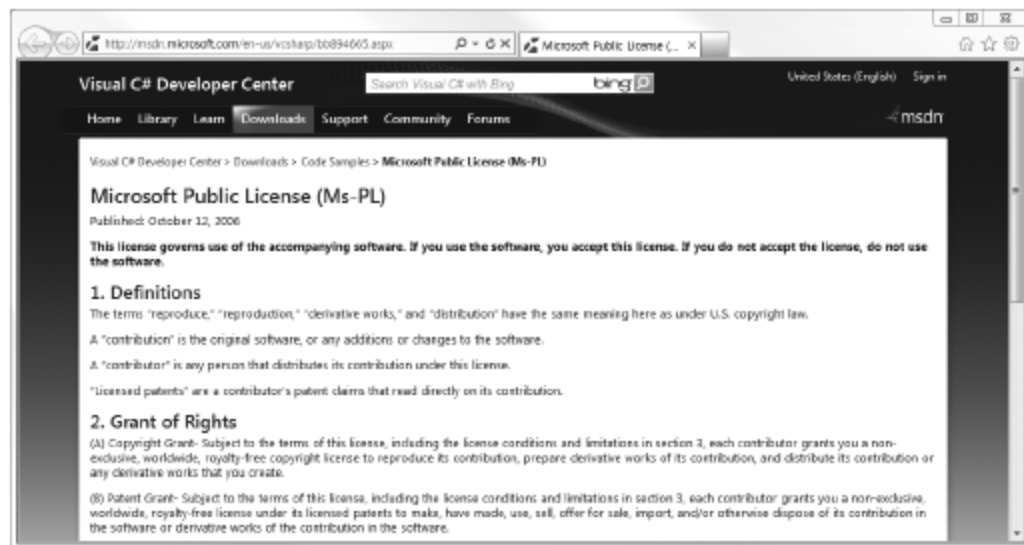


그림 27-4 Dynamic.cs 파일을 Controllers 폴더로 이동



❖ jqGrid 플러그인

- jqGrid 플러그인의 공식 사이트 : <http://www.trirand.com/blog/>
 - css 폴더: jqGrid 플러그인의 스타일 시트
 - js 폴더: 다운로드할 때 선택한 파일
 - plugins 폴더: jqGrid 플러그인과 관련된 플러그인
 - src 폴더: jqGrid 플러그인의 소스

그림 27-5 jQuery Grid Plugin(<http://www.trirand.com/blog/>)

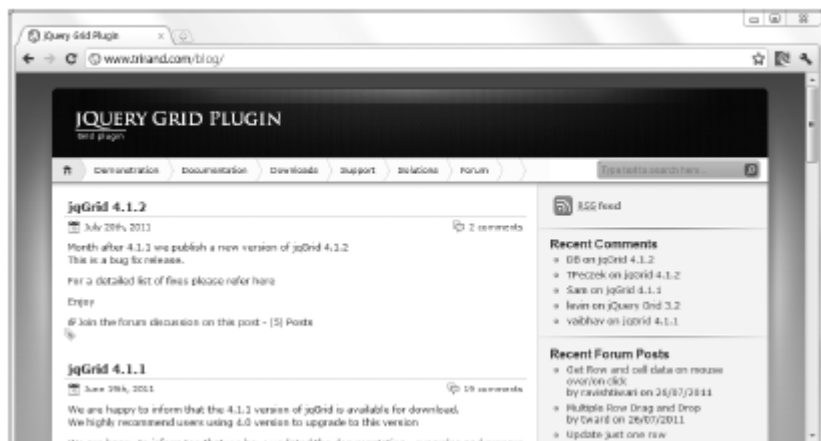


그림 27-6 전체적인 프로젝트 구성

- Content
 - ▶ images
 - jquery-ui-1.8.14.custom.css
 - ui.jqgrid.css
- Controllers
 - Dynamic.cs
 - HomeController.cs
- Models
 - DataClasses.dhtml
 - DataClasses.dhtml.layout
 - DataClasses.designer.cs
- Scripts
 - i18n
 - plugins
 - jquery-ui-1.8.14.custom.min.js
 - jquery.jqGrid.min.js



❖ 테이블 Books

- 데이터베이스 파일을 만들고 그림 27-7처럼 테이블 디자인
- Id에 기본키 주고 Id 설정 입력
- 데이터를 20개 정도 직접 입력
- LINQ To SQL 클래스를 만들고 그림 27-8처럼 Books 테이블 드래그

그림 27-7 테이블 디자인

필드 이름	데이터 형식	Null 허용
Id	int	<input type="checkbox"/>
Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
Author	nvarchar(MAX)	<input checked="" type="checkbox"/>
Publisher	nvarchar(MAX)	<input checked="" type="checkbox"/>
Isbn	nvarchar(MAX)	<input checked="" type="checkbox"/>
Page	int	<input checked="" type="checkbox"/>
PublishDate	date	<input checked="" type="checkbox"/>

그림 27-8 LINQ To SQL 구성



27.4 jqGrid 플러그인을 위한 웹 서비스 구성

❖ jqGrid 플러그인을 사용하기 위한 조건

- 웹 서비스가 다음 다섯 가지 기능 수행해야 함
 - 데이터 제공
 - 데이터 검색
 - 데이터 추가
 - 데이터 삭제
 - 데이터 수정
- jqGrid 플러그인은 두 개의 웹 서비스와만 Ajax 통신을 할 수 있음
 - 다섯 개의 기능을 두 가지 웹 서비스에 모아야

그림 27-9 jqGrid 플러그인을 위한 웹 서비스



❖ HomeController 작성

- 액션 GetBooks는 데이터와 검색 제공하는 웹 서비스
- 액션 EditBooks는 데이터 추가, 삭제, 수정하는 웹 서비스

코드 27-1 HomeController.cs: 액션 GetBooks와 액션 EditBooks 생성

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

    public ActionResult GetBooks()
    {
    }

    public ActionResult EditBooks()
    {
    }
}
```



❖ 액션 GetBooks

- 데이터베이스에서 데이터 가져와 XML 문서 만들어 제공
 - 그리드 화면에 출력할 데이터만 제공
 - Ex) ‘jQuery’ 라고 검색 -> 데이터베이스에서 jQuery가 입력돼 있는 내용을 찾고, 해당 데이터만 XML 문서로 만들어 제공
 - 검색과 정렬 기능을 쉽게 구현하기 위해 Dynamic LINQ To SQL 추가
 - HomeController.cs 파일에 추가
 - » using System.Linq.Dynamic;

```
<?xml version="1.0" encoding="utf-8"?>
<rows>
  <page>현재 페이지</page>
  <total>총 페이지 수</total>
  <records>총 데이터 수</records>
  <row>
    <cell>데이터</cell>
    ...
    <cell>데이터</cell>
  </row>
  ...
  <row>
    <cell>데이터</cell>
    ...
    <cell>데이터</cell>
  </row>
</rows>
```



❖ jqGrid 플러그인의 요청 매개 변수

표 27-1 jqGrid 플러그인의 요청 매개 변수

매개 변수 이름	자료형	설명
page	숫자	그리드에 출력하려는 페이지 번호를 의미합니다.
rows	숫자	그리드에 출력하려는 행 개수를 의미합니다.
sidx	문자열	정렬하려는 열을 의미합니다.
sord	문자열	정렬 방식을 의미합니다(asc 또는 desc).
_search	불리언	검색을 사용하는지 나타냅니다.
searchField	문자열	검색 대상 열을 의미합니다.
searchOper	문자열	검색 방식을 의미합니다.
searchString	문자열	검색 키워드입니다.



❖ 액션 GetBooks의 진행 순서

- 매개 변수 처리
- 데이터베이스에서 데이터 가져옴
- 검색 수행
- XML 문서 만드는 데 사용하는 변수 선언
- XML 문서 생성
- 리턴



❖ 매개변수 처리

- 검색을 제외한 매개 변수만 사용
- jqGrid에서 정렬을 선택하지 않았을 때
 - 빈 문자열이 매개 변수 sidx로 들어감 → 에러 발생 → 막기 위해 1로 처리

코드 27-3 HomeController.cs: 데이터 정렬

```
public ActionResult GetBooks(int page, int rows, string sidx, string sord)
{
    // 매개 변수를 처리합니다.
    if (sidx == "") { sidx = "1"; }

    // 데이터베이스에서 데이터를 가져옵니다.
    DataClassesDB db = new DataClassesDB();
    var Query = db.Books.OrderBy(sidx + " " + sord);
}
```



❖ 페이징과 문서 생성

- 가져온 데이터를 바탕으로 XML 문서 생성 후 리턴
- 검색할 때는 매개 변수 `_search`가 `true`로 전달 → 검색 수행

코드 27-4 HomeController.cs: 페이징과 XML 문서 생성

```
public ActionResult GetBooks(int page, int rows, string sidx, string sord)
{
    // 매개 변수를 처리합니다.
    if (sidx == "") { sidx = "1"; }

    // 데이터베이스에서 데이터를 가져옵니다.
    DataClassesDB db = new DataClassesDB();
    var Query = db.Books.OrderBy(sidx + " " + sord);

    // 검색을 수행합니다.
    if (Request["_search"]=="true")
    {

    }

    // XML을 만드는데 사용하는 변수를 선언합니다.
    var totalRecords = Query.Count();
    var totalPages = Math.Ceiling((double)totalRecords / rows);
    var start = rows * page - rows;
    var books = Query.Skip(start).Take(rows);
```



❖ 페이징과 문서 생성 (2)

```
// XML을 만듭니다.
var willReturn = "";
willReturn += "<rows>";
willReturn += "    <page>" + page + "</page>";
willReturn += "    <total>" + totalPages + "</total>";
willReturn += "    <records>" + totalRecords + "</records>";

foreach (var item in books)
{
    willReturn += "<row>";
    willReturn += "    <cell>" + item.Id + "</cell>";
    willReturn += "    <cell>" + item.Name + "</cell>";
    willReturn += "    <cell>" + item.Author + "</cell>";
    willReturn += "    <cell>" + item.Publisher + "</cell>";
    willReturn += "    <cell>" + item.Isbn + "</cell>";
    willReturn += "    <cell>" + item.Page + "</cell>";
    willReturn += "    <cell>" + item.PublishDate + "</cell>";
    willReturn += "</row>";
}
willReturn += "</rows>";

// 리턴합니다.
return Content(willReturn, "text/xml");
}
```



❖ head 태그 구성

- 웹 서비스를 사용해 기본 그리드 작성
- jqGrid 플러그인 사용

코드 27-5 Index.cshtml: head 태그 구성

```
<head>
  <title>Index</title>
  <link href="../../../Content/jquery-ui-1.8.14.custom.css" rel="Stylesheet"/>
  <link href="../../../Content/ui.jqgrid.css" rel="Stylesheet"/>

  <script src="http://code.jquery.com/jquery-1.7.js"X/script>
  <script src="../../../Scripts/i18n/grid.locale-en.js"X/script>
  <script src="../../../Scripts/jquery.jqGrid.min.js"X/script>
  <script>

  </script>
</head>
```



❖ Body 태그 구성

- jqGrid 플러그인을 사용하려면 table 태그와 div 태그 두 개가 필요
- 그림 27-10에서 그리드 자체가 table 태그

코드 27-6 Index.cshtml: body 태그 구성

```
<body>
  <table id="grid"></table>
  <div id="pager"></div>
</body>
```

그림 27-10 table 태그와 div 태그의 용도

Books					
ID	Name	Author	Publisher	ISBN	Page
11	HEAD FIRST DESIGN PATTERNS	데릭 프리먼	한빛미디어	8979143400	672
12	뇌를 자극하는 PHP 프로그래밍	조명진	한빛미디어	8979146582	688

+

↶

↷

↶

↷

Page 1 of 2

10

▼



❖ jqGrid 플러그인 사용

- jqGrid() 메서드의 첫 번째 매개 변수에는 옵션 입력
 - 설정해야 설정해야 하는 옵션

표 27-2 jqGrid 플러그인의 기본 옵션

옵션 속성 이름	설명
url	Ajax를 수행할 대상 URL을 지정합니다.
pager	도구 모음이 될 div 태그를 지정합니다.
caption	그리드의 제목을 지정합니다.
height	그리드의 높이를 지정합니다.
rowNum	한 화면에 표시할 행 개수를 지정합니다.
rowList	rowNum을 선택할 수 있는 옵션을 지정합니다.
colNames	열 이름을 지정합니다.
colModel	열을 어떤 형태로 생성할지 지정합니다.



❖ 그리드 출력 코드

코드 27-8 Index.cshtml: 옵션 입력

```
<script>
$(document).ready(function () {
    $('#grid').jqGrid({
        url: '/Home/GetBooks',
        pager: '#pager',
        caption: 'Books',
        height: '100%',
        rowNum: 10,
        rowList: [10, 20, 30],
        colNames: ['ID', 'Name', 'Author', 'Publisher', 'ISBN', 'Page',
            'PublishDate'],
        colModel: [
            { name: 'Id', index: 'Id', width: 30 },
            { name: 'Name', index: 'Name', width: 200 },
            { name: 'Author', index: 'Author', width: 90 },
            { name: 'Publisher', index: 'Publisher', width: 80 },
            { name: 'Isbn', index: 'Isbn', width: 80 },
            { name: 'Page', index: 'Page', width: 40 },
            { name: 'PublishDate', index: 'PublishDate', width: 150 }
        ]
    });
});
</script>
```



27.6 기본 그리드 만들기

❖ 실행 결과

그림 27-11 설명

Books						
ID	Name	Author	Publisher	ISBN	Page	PublishDate
1	만드roid 프로그래밍 정국	김상형	한빛미디어	8979148186	2420	2011-05-25 오전 12:00:00
2	만드roid의 모든 것 도서와 포털	고형필	한빛미디어	8979147708	792	2011-06-20 오전 12:00:00
3	100만 방문자와 소통하는 파워블로그 만	윤상진 외 5명	한빛미디어	8979147953	396	2011-05-06 오전 12:00:00
4	Head First Java	케미시 시메리	한빛미디어	8979143710	720	2007-11-09 오전 12:00:00
5	회사에서 바로 통하는 일러스트레이터 디	박해진	한빛미디어	8979148240	548	2011-05-01 오전 12:00:00
6	ITQ 파워 포인트 2007	한빛OA연구회	한빛미디어	8979148119	184	2011-02-25 오전 12:00:00
7	ITQ 엑셀 2007	한빛OA연구회	한빛미디어	8979148003	200	2011-02-03 오전 12:00:00
8	웹 표준 가이드	존 앨싱	한빛미디어	8979147945	464	2010-11-30 오전 12:00:00
9	사전처럼 바로 찾아 쓰는 JQUERY	B.M. 히와니	한빛미디어	8979148054	474	2011-02-25 오전 12:00:00
10	한 번에 배우는 HTML5 지바스크립트	지닌 미디어	한빛미디어	8979148267	380	2011-04-30 오전 12:00:00

Page 1 of 2 10



❖ colModel 옵션 속성

- 열을 어떤 형식으로 지정할 것인지 설정
 - colModel 옵션 속성은 배열
 - 각 배열 요소는 객체로 구성
 - 객체의 속성으로 지정할 수 있는 속성이 34개
 - 자주 쓰이는 객체 속성

표 27-3 자주 사용하는 colModel의 속성

옵션 속성 이름	설명
align	데이터의 정렬 방식을 지정합니다(left, center, right).
editable	수정 가능 여부를 지정합니다.
editoptions	수정 옵션을 지정합니다.
edittype	수정 형태를 지정합니다.
index	매개 변수 sidx를 정의합니다.
name	jqGrid 플러그인이 열을 구분하기 위한 고유 이름입니다.
search	검색 가능 여부를 지정합니다.
searchoptions	검색 옵션을 지정합니다.
sortable	정렬 가능의 유무를 지정합니다.
width	열의 너비를 지정합니다.



❖ 웹 서비스 완성

- 예제 27-9] 처럼 navGrid() 메서드 사용해야 도구 모음 볼 수 있음

코드 27-9 Index.cshtml: 도구 모음 생성

```
$('#grid').jqGrid({
    /* 생략 */
}).navGrid('#pager', {
    search: true,
    edit: true,
    add: true,
    del: true
});
```

그림 27-12 검색

Search...				Page
ID	equal			6 2420
				8 792
				3 396
4	Head First Java	캐시미어	한빛미디어	89791437 10 720
5	회사에서 바로 통하는 일래스트레이터 디렉토리	한빛미디어	한빛미디어	89791482 40 548
6	ITQ 파워포인트 2007	한빛OA연구회	한빛미디어	89791481 19 184
7	ITQ 엑셀 2007	한빛OA연구회	한빛미디어	89791480 03 200
8	원 포인트 가이드	존 웹서	한빛미디어	89791479 45 464
9	서전지컬 바로 읽어 쓰는 JQUERY	B.M. 최오리	한빛미디어	89791480 54 474
10	한 번에 배우는 HTML5 자바스크립트	지남 미디어	한빛미디어	89791482 67 380



❖ 검색 기능 구현

- searchField, searchOper, searchString라는 매개 변수로 전달
- 액션 GetBooks의 검색 수행 부분
 - 변수 searchField, searchString 초기화
 - Where() 메서드 사용해 검색

코드 27-10 HomeController.cs: 검색 구현

```
// 검색을 수행합니다.
if (Request["_search"]=="true")
{
    // 검색이라면 검색 키워드에 해당하는 내용만 가져옵니다.
    var searchField = Request["searchField"];
    var searchString = Request["searchString"];
    var query = "";
    if (searchField == "Id")
    {
        Query = Query.Where(searchField + "==" + searchString);
    }
    else
    {
        Query = Query.Where(searchField + ".Contains(@0)", searchString);
    }
}
```



❖ jqGrid 플러그인을 위한 웹 서비스의 구현 완성

- 데이터 추가하거나 삭제, 수정하는 웹 서비스를 하나만 선택
 - 추가, 삭제, 수정 구분하는 매개 변수 oper 전달
 - 기능 구분해 구현 가능

코드 27-11 HomeController.cs: 액션 EditBooks 기능 분화

코드 27-12 HomeController.cs: 액션 EditBooks 완성

```
public ActionResult EditBooks(string oper)
{
    // 데이터베이스에서 데이터를 가져옵니다.
    DataClassesDB db = new DataClassesDB();

    // 각각에 맞게 처리합니다.
    if (oper == "add")
    {
        // 책을 추가합니다.
        Books books = new Books();
        books.Name = Request["Name"];
        books.Author = Request["Author"];
        books.Publisher = Request["Publisher"];
        books.Isbn = Request["Isbn"];
        books.Page = Request["Page"];
        books.PublishDate = DateTime.Now;

        db.Books.InsertOnSubmit(books);
        db.SubmitChanges();
    }
}
```



❖ jqGrid 플러그인을 위한 웹 서비스의 구현 완성 (2)

```
else if (oper == "edit")
{
    // 책을 변경합니다.
    var id = int.Parse(Request["Id"]);
    var willUpdate = db.Books.Single(x => x.Id == id);
    willUpdate.Name = Request["Name"];
    willUpdate.Author = Request["Author"];
    willUpdate.Publisher = Request["Publisher"];
    willUpdate.Isbn = Request["Isbn"];
    willUpdate.Page = Request["Page"];
    willUpdate.PublishDate = DateTime.Now;

    db.SubmitChanges();
}
else if (oper == "del")
{
    // 책을 삭제합니다.
    var id = int.Parse(Request["Id"]);
    var willDelete = db.Books.Single(x => x.Id == id);
    db.Books.DeleteOnSubmit(willDelete);
}
return Content("SUCCESS");
}
```



27.9 그리드에 추가, 수정, 삭제 기능 추가

❖ 추가, 삭제, 수정이 가능하게 하려면?

- colModel 객체에 editable 속성 true로 지정
- 기본키는 처음 데이터베이스를 만들 때 자동으로 증가하도록
 - 수정 금지

코드 27-13 Index.cshhtml: colModel 옵션 속성 지정

```
colModel: [  
  { name: 'Id', index: 'Id', width: 30 },  
  { name: 'Name', index: 'Name', width: 200, editable: true },  
  { name: 'Author', index: 'Author', width: 90, editable: true },  
  { name: 'Publisher', index: 'Publisher', width: 80, editable: true },  
  { name: 'Isbn', index: 'Isbn', width: 80, editable: true },  
  { name: 'Page', index: 'Page', width: 40, editable: true },  
  { name: 'PublishDate', index: 'PublishDate', width: 150, editable: true }  
]
```



❖ customDialog 선언

- 데이터를 수정하거나 삭제할 때는 기본 키 웹 서비스에 전달해야
 - 수정할 때 editable 속성이 true인 속성만 웹 서비스로 전달
 - 삭제 시에는 현재 그리드에서 대상의 위치 정보만 웹 서비스로 전달

코드 27-14 Index.cshhtml: 사용자 지정 다이얼로그 생성

```
<script>
$(document).ready(function?() {
    // 변수를 선언합니다.
    var customDialog = {
        onclickSubmit: function (params) {
            var selectedRow = $('#grid').getGridParam("selrow");
            rowData = $('#grid').getRowData(selectedRow);
            return { Id: rowData.Id };
        }
    };

    // 그리드를 생성합니다.
    $('#grid').jqGrid({
        /* 생략 */
    }).navGrid('#pager', {
        search: true,
        edit: true,
        add: true,
        del: true
    }, customDialog, {}, customDialog);
});
</script>
```



27.9 그리드에 추가, 수정, 삭제 기능 추가

❖ navGrid() 메서드의 완전한 형태

- 예제에서는 수정하거나 삭제할 때 customDialog 객체 사용 가능
 - navGrid('#pager', { options }, edit, add, del, search, view);
- customDialog 객체는 submit 이벤트가 발생할 때 Id를 함께 넘김
 - 선택된 row를 가져와 객체로 만들어 리턴

```
// 변수를 선언합니다.  
var customDialog = {  
    onclickSubmit: function (params) {  
        var selectedRow = $("#grid").getGridParam("selrow");  
        rowData = $("#grid").getRowData(selectedRow);  
        return { Id: rowData.Id };  
    }  
};
```

그림 27-13 설명

The screenshot shows a web application with a 'Add Record' dialog box on the left and a table of books on the right. The dialog box has fields for Name, Author, Publisher, ISBN, and Page, with 'Submit' and 'Cancel' buttons. The table has columns for Name, Author, Publisher, ISBN, and Page, and contains several rows of book data.

그림 27-14 설명

The screenshot shows a web application with a table of books. The table has columns for Name, Author, Publisher, ISBN, and Page, and contains several rows of book data. The table is titled 'Books' and has a 'Page' column.

ID	Name	Author	Publisher	ISBN	Page
11	HEAD FIRST DESIGN PATTERNS	헤릭 프리먼	한빛미디어	8979143400	672
12	뇌를 자극하는 PHP 프로그래밍	조영진	한빛미디어	8979146582	688



27.9 그리드에 추가, 수정, 삭제 기능 추가

❖ colModel 객체의 edittype 속성

표 27-4 jQuery Grid의 edittype 속성

문자열	설명
text	<code><input type="text" /></code>
textarea	<code><textarea>/></code>
select	<code><select>/></code>
checkbox	<code><input type="checkbox" /></code>
password	<code><input type="password" /></code>
button	<code><input type="button" /></code>
image	<code><input type="image" /></code>
file	<code><input type="file" /></code>



❖ colModel 객체의 edittype 속성 사용 예제

코드 27-15 Index.cshtml: edittype 속성

```
colModel: [  
  { name: 'Id', index: 'Id', width: 30 },  
  { name: 'Name', index: 'Name', width: 200, editable: true, edittype: 'text' },  
  { name: 'Author', index: 'Author', width: 90, editable: true, edittype: 'textarea' },  
  { name: 'Publisher', index: 'Publisher', width: 80, editable: true, edittype:  
    'password' },  
  { name: 'Isbn', index: 'Isbn', width: 80, editable: true, edittype: 'checkbox' },  
  { name: 'Page', index: 'Page', width: 40, editable: true, edittype: 'file' },  
]
```

그림 27-15 설명



The screenshot shows a web form titled "Add Record" with a close button (X). It contains the following fields:

- Name: A single-line text input field.
- Author: A multi-line text area.
- Publisher: A single-line password input field.
- Isbn: A checkbox.
- Page: A file input field with a button labeled "파일 선택" (Select File) next to it.



❖ 그리드 이벤트

- jqGrid 플러그인은 총 18개의 이벤트 지원
 - 이벤트를 연결하려면, jqGrid() 메서드의 매개 변수 옵션에 이벤트 입력

코드 27-16 Index.cshtml: 이벤트 연결

```
$('#grid').jqGrid({  
  url: '/Home/GetBooks',  
  /*  
  생략  
  */  
  colModel: [  
    /* 생략 */  
  ],  
  ondblClickRow: function (rowid, row, col, event) {  
    $('#grid').editGridRow(rowid);  
  }  
})
```





Thank You !

모던 웹을 위한 Javascript jQuery 입문