

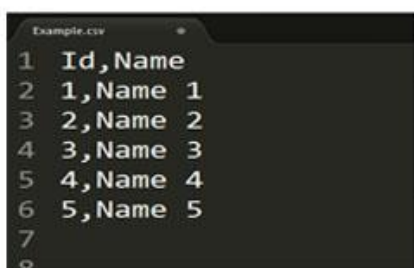
در این مقاله قصد داریم در مورد خواندن فایل CSV و چگونگی خواندن آن با کمک CSV لinq و بدون کمک لinq در CSV فایل را نمایش دهیم .

نحوه خواندن فایل در Asp.Net Mvc

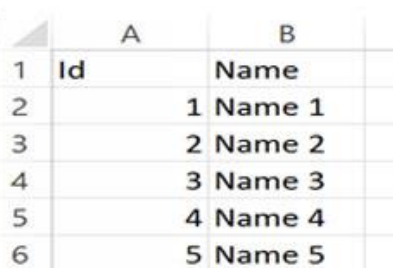
برنامه نویسی را از برنامه نویسان حرفه ای بیاموزید

In a text Editor

In Excel



```
Example.csv
1 Id,Name
2 1,Name 1
3 2,Name 2
4 3,Name 3
5 4,Name 4
6 5,Name 5
7
8
```



	A	B
1	Id	Name
2		1 Name 1
3		2 Name 2
4		3 Name 3
5		4 Name 4
6		5 Name 5

mve



دو روش برای خواندن داده ها از فایل CSV وجود دارد:

۱- خواندن فایل CSV بدون هیچ کتابخانه خاص

۲- خواندن فایل CSV با استفاده از linq to csv

در مورد فایل CSV هر خط از فایل یک رکورد است و هر رکورد یک سری اطلاعات دارد.

به عنوان مثال یک نمونه از فایل CSV:

```
Names,Dates
Name 1,12/12/2015
Name 2,12/11/2015
```

کاربر باید بین دو روش استفاده از CSV یکی را انتخاب نماید یا از کتابخانه CSV استفاده کنند و یا اینکه از کتابخانه CSV استفاده نکند.

وقتی که بر روی دکمه کلیک می نمایید یک دکمه برای ارسال یک درخواست http به سرور است که به تجزیه و تحلیل فایل CSV می پردازد.

در این نرم افزار رکورد اول به عنوان هدر سایت قرار می گیرد.

Names	Dates
Name 1	12/12/2015
Name 2	12/11/2015
Name 3	12/10/2015
Name 4	11/10/2015
Name 5	09/05/2015

← First line



در زیر برای پیاده سازی متد از کد زیر استفاده می نمایم:

```
public IEnumerable<Person> Method1(string fileSrc, char sep)
{ //Without using LinqToCsv lib
  var list = new List<Person>();
  var reader = new StreamReader( fileSrc );
  bool header = true;
  while (!reader.EndOfStream)
  {
    //get the first line
    var line = reader.ReadLine();
    //specify the separator line
    //Split : string is separated by a delimiter to form a table as result
    var values = line.Split(sep);
    Person person = new Person();
    person.Name = values[0];
    person.Date = values[1];
    if(header == true)
    { //escape first line
      header = false;
    }
    else
    {
      list.Add(person);
    }
  }
}
```

```
return list.AsEnumerable();  
}
```

ویرایش فایل csv با استفاده از یک ویرایشگر

زمانی که برنامه با خطا مواجه شود:

Names	Dates
;;1	12/12/2015
Na,,me; 2	12/11/2015
Name;; 3	12/10/2015
N;ame; 4	11/10/2015
N,a;m;e; 5	09/05/2015

Names	Dates
"	
"Na,,me	2"
"Name	
"N	ame
"N,a	m



برای کار کردن حتما باید dll مربوطه را به آن اضافه نمایید :

Microsoft.VisualBasic.FileIO.TextFieldParser

اول از همه باید TextFieldParse اضافه کنیم، پس حتما کتابخانه Microsoft.VisualBasic را اضافه نمایید.



```
public IEnumerable<Person> Method1(string fileSrc, string sep)
{ //Without using LinqToCsv lib
  var list = new List<Person>();
  bool header = true;
  using (TextFieldParser parser = new TextFieldParser(fileSrc))
  {
    parser.TextFieldType = Microsoft.VisualBasic.FileIO.FieldType.Delimited;
    //specify the delimiter
    parser.SetDelimiters(sep);
    while (!parser.EndOfData)
    {
      //read Fileds
      string[] fields = parser.ReadFields();
      //Processing row
      if (header == true)
      { //escape first line
        header = false;
      }
      else
      {
        Person person = new Person();
        person.Name = fields[0];
        person.Date = fields[1];
        list.Add(person);
      }
    }
  }
}
```

```
}  
}  
return list.AsEnumerable();  
}
```

مرجع تخصصی برنامه نویسان

انتخاب متد :

متد اول : بدون کتابخانه LinqTo csv

متد دوم : کتابخانه LinqTo csv

Start parsing

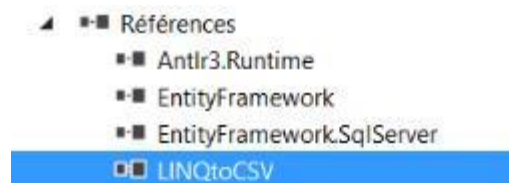
نتیجه :

Dates	Names
12/12/2015	Name 1
12/11/2015	Name 2
12/10/2015	Name 3
11/10/2015	Name 4
09/05/2015	Name 5



حالا با استفاده از کتابخانه Linq To csv

اول باید کتابخانه را اضافه نمایم.



پیاده سازی:

```
public IEnumerable<Person> Method2(string fileSrc, char sep)
{ //Using LinqToCsv lib
  CsvFileDescription inputFileDescription = new CsvFileDescription
  {
    SeparatorChar = sep, //specify the separator line
    FirstLineHasColumnNames = true //Header is in the first line
  };
  CsvContext csvContext = new CsvContext();
  IEnumerable<Person> list =
    csvContext.Read<Person>(fileSrc, inputFil, Description);
  // Data is now available via variable list.
  return list;
}
```

حالا باید کد های html را قرار دهید:

```
<div class="row">
  <div class="col-lg-12">
    <form class="form" method="POST"
      action="/Home/Index">
```

```

<div class="form-group">
  <label>Choose a method :</label>
  <br>
  <input type="radio" name="method"
  value="0" checked />
  Method 1 : without LinqToCSV lib<br>
  <input type="radio" name="method"
  value="1" /> Method 2 : LinqToCSV lib
</div>
<div class="form-group">
  <input type="submit"
  class="form-control btn btn-default"
  value="Start parsing" />
</div>
</form>
</div>
</div>
<hr>
<div class="row">
  <div class="col-lg-12">
    <label>Parsing result :</label>
  </div>
</div>
<div class="row">
  <div class="col-sm-4">
    <table class="table">
      @if (@Model != null)
      {
        @Html.Raw("<thead><th>Names</th>
        <th>Dates</th></thead><tbody>");
        foreach (var elem in @Model as IEnumerable
        <WebApplicationReadingCSVFile.Models.Person>)
        {
          @Html.Raw("<tr ><td>" +
          @elem.Name + "</td><td>" +
          @elem.Date + "</td></tr>");
        }
        @Html.Raw("</tbody>");
      }
    </table>
  </div>
</div>

```

```
</div>  
<div class="col-sm-8"></div>  
</div>
```

نوشتن کد های مربوط به #asp.net mvc :

```
public class Person  
{  
    //CsvColumn is key word used by LinqToCSV  
    //CsvColumn : specify the Name of Header and the index of column  
    [CsvColumn(Name = "Names", FieldIndex = 1)]  
    public string Name { get; set; }  
    [CsvColumn(Name = "Dates", FieldIndex = 2)]  
    public string Date { get; set; }  
}
```

مرحله ی بعدی نوشتن متد action result است:

```
public ActionResult Index()  
{  
    //get the selected radio value  
    string method = Request.Params["method"];  
    char sep = ',';  
    //initialize list  
    IEnumerable<Person> list = new List<Person>();  
    if (method != null && method != "")  
    {  
        //Get Server Path of CSV file  
        string fileSrc = Server.MapPath("~/Content/Files/CsvFile.csv");  
        //Build Result  
        if (method == "0")  
        {  
            list = Method1(fileSrc, sep);  
        }  
        else{  
            list = Method2(fileSrc, sep);  
        }  
        //return the list as Enumerable  
    }  
}
```



```
return View(list);  
}
```

خروجی:

مرجع تخصصی برنامه نویسان

انتخاب متد :

متد اول : بدون کتابخانه LinqTo csv

متد دوم : کتابخانه LinqTo csv

Start parsing

نتیجه :

Dates	Names
12/12/2015	Name 1
12/11/2015	Name 2
12/10/2015	Name 3
11/10/2015	Name 4
09/05/2015	Name 5

