

The results are in! See what nearly 90,000 developers picked as their most loved, dreaded, and desired coding languages and more in the 2019 Developer Survey.

LINQ query on a DataTable

[Ask Question](#)

951



194

I'm trying to perform a LINQ query on a DataTable object and bizarrely I am finding that performing such queries on DataTables is not straightforward. For example:

```
var results = from myRow in myDataTable
where results.Field("RowNo") == 1
select results;
```

This is not allowed. How do I get something like this working?

I'm amazed that LINQ queries are not allowed on DataTables!

c#

.net

linq

datatable

.net-3.5

edited Jul 4 '14 at 20:44



[Peter Mortensen](#)

13.9k 19 87 113

asked Aug 14 '08 at 10:08



[Calanus](#)

11.3k 22 71 108

- 6 You want what's known as [LINQ to DataSet](#). That link will take you to the first in a series of posts introducing it on the ADO.NET team blog. – [Matt Hamilton](#) Aug 14 '08 at 10:11
- 3 You can find more LINQ/Lambda example from [webmingle.blogspot.com/2010_09_01_archive.html](#) – user562221 Feb 17 '11 at 19:18

21 Answers

Home

PUBLIC

 Stack Overflow

Tags

Users

Jobs

Teams

Q&A for work



[Learn More](#)

1196



You can't query against the `DataTable` 's `Rows` collection, since `DataRowCollection` doesn't implement `IEnumerable<T>` . You need to use the `AsEnumerable()` extension for `DataTable` . Like so:

```
var results = from myRow in myDataTable.AsEnumerable()
where myRow.Field<int>("RowNo") == 1
select myRow;
```

And as Keith says, you'll need to add a reference to [System.Data.DataSetExtensions](#)

`AsEnumerable()` returns `IEnumerable<DataRow>` . If you need to convert `IEnumerable<DataRow>` to a `DataTable` , use the `CopyToDataTable()` extension.

Below is query with Lambda Expression,

```
var result = myDataTable
.AsEnumerable()
.Where(myRow => myRow.Field<int>("RowNo") == 1);
```

edited Feb 19 at 15:04



[Harald Coppoolse](#)

13k 1 30 65

answered Aug 14 '08 at 19:45



Collin K

12.8k 1 19 18

-
- 7 VB Version: Dim results = From myRow In myDataTable.AsEnumerable
_ Where myRow.Field("RowNo") = 1 _ Select myRow – [Jeff](#) Jul 29 '09 at 20:46
-
- 14 I already had a reference to the dll mentioned, but was missing using
System.Data; – [Luke Duddridge](#) May 31 '11 at 10:37
-
- 4 VB Version needs to insert (Of String) between myRow.Field and
("RowNo"). That part should read: myRow.Field(Of String)("RowNo") = 1
- Reference @Cros comment. – [yougotiger](#) Jun 18 '12 at 22:26
-
- 6 this solution is needlessly complicated. Use myDataTable.Rows instead
as @JoelFan suggested. – [The Conspiracy](#) Jun 25 '14 at 18:46
-
- 7 @Markus Just to clarify, the reason that @JoelFan's solution works with
myDataTable.Rows is because the myRow variable is explicitly cast to
DataRow . When it is compiled, that query is rewritten to
myDataTable.Rows.Cast<DataRow>().Where(myRow =>
(int)myRow["RowNo"] == 1) . Personally, I don't find the call to
AsEnumerable() any more complicated than the call to Cast<DataRow>
() . As far as I know, the performance is the same, so it's just a matter
of preference. – [Collin K](#) Jun 26 '14 at 16:29
-



you can try this, but you must be sure the type of values for each
Column

4



```
List<MyClass> result = myDataTable.AsEnumerable().Select(x=> new MyC.  
    Property1 = (string)x.Field<string>("ColumnName1"),  
    Property2 = (int)x.Field<int>("ColumnName2"),
```

```
Property3 = (bool)x.Field<bool>("ColumnName3"),
});
```

edited Sep 25 '18 at 14:33

answered Feb 1 '18 at 21:43



Gabriel Martinez Bustos

71 6

Has the world gone mad? Whats wrong with sql? DataRow[] drs = dt.Select("id=1"); Maybe this is too easy. – Programnik Sep 23 '18 at 23:17

25

I realize this has been answered a few times over, but just to offer another approach, I like to use the `.Cast<T>()` method, it helps me maintain sanity in seeing the explicit type defined, and deep down I think `.AsEnumerable()` calls it anyways:

```
var results = from myRow in myDataTable.Rows.Cast<DataRow>()
               where myRow.Field<int>("RowNo") == 1 select myRow;
```

or

```
var results = myDataTable.Rows.Cast<DataRow>()
               .FirstOrDefault(x => x.Field<int>("RowNo") == 1);
```

edited Dec 18 '17 at 2:07



Milad Rashidi

826 4 11 29

answered Feb 2 '16 at 21:22



vandsh

654 9 9

4 This works without referencing System.Data.DataSetExtensions. –
[user423430](#) Dec 19 '17 at 15:09



10

Most likely, the classes for the DataSet, DataTable and DataRow are already defined in the solution. If that's the case you won't need the DataSetExtensions reference.



Ex. DataSet class name-> CustomSet, DataRow class name-> CustomTableRow (with defined columns: RowNo, ...)

```
var result = from myRow in myDataTable.Rows.OfType<CustomSet.CustomT;
              where myRow.RowNo == 1
              select myRow;
```

Or (as I prefer)

```
var result = myDataTable.Rows.OfType<CustomSet.CustomTableRow>().Where
myRow.RowNo);
```

edited Dec 15 '17 at 12:16



Paul Zahra

7,469 5 43 59

answered Apr 24 '13 at 17:17



xadriel

181 2 5

17

Try this simple line of query:

```
var result=myDataTable.AsEnumerable().Where(myRow => myRow.Field<int>
```

edited Dec 15 '17 at 12:15



Paul Zahra

7,469 5 43 59

answered Apr 5 '16 at 9:38



Mohit Verma - MSFT

1,501 1 6 17

- 3 I prefer the "Method Chaining" (as you have done here) over the "Query Syntax" (in the accepted answer) simply because this is a basic where-clause that fits on one line and is still very readable. To each their own. – [MikeTeeVee](#) Aug 10 '16 at 7:44

6

Example on how to achieve this provided below:

```
DataSet dataSet = new DataSet(); //Create a dataset
dataSet = _DataEntryDataLayer.ReadResults(); //Call to the dataLayer

//LINQ query on a DataTable
var dataList = dataSet.Tables["DataTable"]
    .AsEnumerable()
    .Select(i => new
    {
        ID = i["ID"],
        Name = i["Name"]
    }).ToList();
```

answered Oct 25 '17 at 16:04



Ryan Gavin

407 1 5 17

Try this...

5

```
SqlCommand cmd = new SqlCommand( "Select * from Employee",con);
SqlDataReader dr = cmd.ExecuteReader( );
DataTable dt = new DataTable( "Employee" );
dt.Load( dr );
var Data = dt.AsEnumerable( );
var names = from emp in Data select emp.Field<String>( dt.Columns[1]
foreach( var name in names )
{
    Console.WriteLine( name );
}
```

edited Apr 20 '17 at 14:17



Tshilidzi Mudau
2,871 23 34

answered Apr 10 '14 at 10:24



Uthaiah
1,103 12 13

27

```
//Create DataTable
DataTable dt= new DataTable();
dt.Columns.AddRange(New DataColumn[]
{
    new DataColumn("ID",typeOf(System.Int32)),
    new DataColumn("Name",typeOf(System.String))
});

//Fill with data

dt.Rows.Add(new Object[]{1,"Test1"});
dt.Rows.Add(new Object[]{2,"Test2"});
```

```
//Now Query DataTable with linq
//To work with linq it should required our source implement IEnumerable
//But DataTable not Implement IEnumerable interface
//So we call DataTable Extension method i.e AsEnumerable() this wil
EnumerableRowCollection<DataRow>
```

```
// Now Query DataTable to find Row whoes ID=1
```

```
DataRow drow = dt.AsEnumerable().Where(p=>p.Field<Int32>(0)==1).Firs
//
```

edited Nov 4 '15 at 22:08



Ayyappan Subramanian

4,416 1 13 35

answered Jan 5 '12 at 8:43



sushil pandey

657 9 9



6



```
IEnumerable<string> result = from myRow in dataTableResult.AsEnumeral
select myRow["server"].ToString() ;
```

answered Aug 4 '15 at 7:32



Iman Abidi

12k 6 58 76



10

This is a simple way that works for me and uses lambda expressions:

```
var results = myDataTable.Select("").FirstOrDefault(x => (int)x["RowI
```


Then if you want a particular value:

```
if(results != null)
    var foo = results["ColName"].ToString()
```

answered Mar 18 '15 at 22:13



Matt Kemp

1,498 18 29

6

In my application I found that using LINQ to Datasets with the AsEnumerable() extension for DataTable as suggested in the answer was extremely slow. If you're interested in optimizing for speed, use James Newtonking's Json.Net library (<http://james.newtonking.com/json/help/index.html>)

```
// Serialize the DataTable to a json string
string serializedTable = JsonConvert.SerializeObject(myDataTable);
Jarray dataRows = Jarray.Parse(serializedTable);

// Run the LINQ query
List<JToken> results = (from row in dataRows
                        where (int) row["ans_key"] == 42
                        select row).ToList();

// If you need the results to be in a DataTable
string jsonResults = JsonConvert.SerializeObject(results);
DataTable resultsTable = JsonConvert.DeserializeObject<DataTable>(js
```

answered Oct 14 '14 at 17:51



LandedGently

563 1 7 16

I doubt this is faster, in the general cases. It has the overhead of two serialization, one deserialization and one parsing operations. Regardless,

I downvoted because it is not concise, i.e. the serialization/deserialization doesn't make clear that the intent is to filter a list. – [an phu](#) Aug 7 '15 at 22:09

@an phu, using the .AsEnumerable extension method creates a collection of heavyweight `System.Data.DataRow` objects. The serialized and parsed data table creates lightweight data consisting only of the column names and values of each row. When the query runs, it will load the data into memory, which for a large dataset may involve swapping. Sometimes, the overhead of several operations is less than the overhead of copying large amounts of data in and out of memory. – [LandedGently](#) Aug 10 '15 at 19:42

[Using LINQ to manipulate data in DataSet/DataTable](#)

27

```
var results = from myRow in tblCurrentStock.AsEnumerable()
              where myRow.Field<string>
("item_name").ToUpper().StartsWith(tbSearchItem.Text.ToUpper())
              select myRow;
DataView view = results.AsDataView();
```

edited Feb 11 '14 at 4:10



[Max](#)

14k 3 29 46

answered Jul 13 '11 at 11:21



[Salim](#)

279 3 2

- 1 The AsDataView doesn't appear in Intellisense for me. I included using `System.Data.Linq` and using `System.Linq` but still it's not working. Do you know what am I missing? Thanks in advance. – [Naomi](#) May 9 '13 at 19:27



@Naomi It comes from `System.Data.DataSetExtensions` . – [Louis](#) Feb 19 '14 at 16:51

For VB.NET The code will look like this:

6

```
Dim results = From myRow In myDataTable
Where myRow.Field(Of Int32)("RowNo") = 1 Select myRow
```

edited Feb 11 '14 at 4:07



[Klaster_1](#)

8,111 5 50 53

answered Oct 17 '12 at 16:04



[Abdul Saboor](#)

2,776 1 24 23

9

```
var results = from myRow in myDataTable
where results.Field<Int32>("RowNo") == 1
select results;
```

edited Feb 1 '14 at 12:16



[falsetru](#)

253k 35 450 443

answered Feb 1 '14 at 11:51



[Vinay](#)

159 2 2

This answer as a lot of issues with it. – [Mr Anderson](#) Aug 15 '16 at 18:05

▲ You can get it work elegant via linq like this:

5

▼

```
from prod in TenMostExpensiveProducts().Tables[0].AsEnumerable()  
where prod.Field<decimal>("UnitPrice") > 62.500M  
select prod
```

Or like dynamic linq this (AsDynamic is called directly on DataSet):

```
TenMostExpensiveProducts().AsDynamic().Where (x => x.UnitPrice > 62.!
```

I prefer the last approach while is is the most flexible. P.S.: Don't forget to connect `System.Data.DataSetExtensions.dll` reference

edited Nov 6 '13 at 9:44

answered Nov 3 '13 at 17:54



AuthorProxy

3,910 3 17 26

Try this

11

▼

```
var row = (from result in dt.AsEnumerable().OrderBy( result => Guid.I  
result).Take(3) ;
```

edited May 18 '12 at 7:51



Marijn

8,836 4 46 72

answered May 18 '12 at 7:15



midhun sankar

119 1 2



38



```
var query = from p in dt.AsEnumerable()
             where p.Field<string>("code") == this.txtCat.Text
             select new
             {
                 name = p.Field<string>("name"),
                 age= p.Field<int>("age")
             };
```

edited Feb 8 '11 at 22:23



Chris

4,761 6 29 51

answered May 23 '10 at 4:03



Ravi

411 4 2

How I use name? For example, `MessageBox.Show(name)` is undefined. – user1372430 Sep 30 '14 at 22:54



122



```
var results = from DataRow myRow in myDataTable.Rows
               where (int)myRow["RowNo"] == 1
               select myRow
```

answered Mar 5 '09 at 2:53



JoelFan

19.5k 28 112 184

1 What about for selecting multiple rows, instead of just row 1? – Adjit Apr 29 '16 at 15:35

1 Just remove the "where" line and you will get all the rows – [JoelFan](#) May 1 '16 at 2:48

Yes, this is how I use to do it, except for replacing `(int)myRow["RowNo"]` with the generic form `myRow.Field<int>("RowNo")` to more conveniently support nullable types. – [Jonas](#) Aug 25 '17 at 14:35

As @ch00k said:

48

```
using System.Data; //needed for the extension methods to work
```

```
...
```

```
var results =
    from myRow in myDataTable.Rows
    where myRow.Field<int>("RowNo") == 1
    select myRow; //select the thing you want, not the collection
```

You also need to add a project reference to
System.Data.DataSetExtensions

answered Aug 14 '08 at 11:07



[Keith](#)

95.1k 59 236 354

You can use LINQ to objects on the Rows collection, like so:

15

```
var results = from myRow in myDataTable.Rows where myRow.Field("RowNo") == 1
               select myRow;
```

answered Aug 14 '08 at 10:11



[David Wengier](#)



8,883 5 33 42

1 Because `DataTable.Rows` does not implement `IEnumerable`, I can't see how this query could compile. – [onedaywhen](#) Jul 11 '16 at 8:45

@onedaywhen I just saw this being done in some code and it does compile. Trying to figure out why right now. – [BVernon](#) Jan 26 '18 at 22:42

didn't work for me. – [TheEsnSiavashi](#) Jul 3 '18 at 21:30



66



It's not that they were deliberately not allowed on `DataTables`, it's just that `DataTables` pre-date the `IQueryable` and generic `IEnumerable` constructs on which Linq queries can be performed.

Both interfaces require some sort type-safety validation. `DataTables` are not strongly typed. This is the same reason why people can't query against an `ArrayList`, for example.

For Linq to work you need to map your results against type-safe objects and query against that instead.

answered Aug 14 '08 at 10:10

[Jon Limjap](#)

79.9k 14 91 146

protected by [Travis J](#) Apr 28 '15 at 22:46

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus does not count](#)).

Would you like to answer one of these [unanswered questions](#) instead?

