

Original source - Sergey Zvezdin: personal blog (<http://blog.zvezdin.com/en/40>)

ADO.NET Data Services 1.5: server driven paging.

We continue to consider new possibilities of ADO.NET Data Services 1.5 and today we will look possibility of compulsory breakdown of data into pages.

Possibility of breakdown of data on pages already was present at first version of Astoria. For these purposes it was possible to construct query by means of Take/Skip operators. Changing values of these parameters it was possible to get paginal access to data from the client. Nevertheless, at the access to a collection without parameters service returned completely all collection. The described behaviour seems quite logical, but it not always is desirable. For example, if in a collection the considerable quantity of objects the access to such collection will by all means entail the raised loading on a server contains. Therefore in version 1.5 the decision on addition of possibility of compulsory breakdown on pages was accepted.

To enable of the described possibility it is necessary to define additional behaviour in configuration of service. In particular SetEntitySetPageSize method which allows to define the size of page for each collection is necessary for us.

```
01 public class WebDataService : DataService<NorthwindEntities>
02 {
03     public static void InitializeService(DataServiceConfiguration config)
04     {
05         config.SetEntitySetAccessRule("*", EntitySetRights.All);
06
07         config.DataServiceBehavior.MaxProtocolVersion = DataServiceProtocolVersion.V2;
08         config.SetEntitySetPageSize("Categories", 5);
09     }
10 }
```

In case of described above service, at the access to Categories collection data will be broken into pages on five objects.

Server Driven Paging



Apparently, for the access to the following page it is necessary to use a \$skiptoken keyword. In this case URI format will look as follows:

`http://.../WebDataService.svc/Categories?$skiptoken=5`

The given functionality also is supported and in a client proxy. At the usual access to a collection the specified number of objects will be given out the client only. For the access to other pages it is necessary to use Skip() method.

```
1 var client = new NorthwindEntities(new Uri(@"..."));
2 var q = from c in client.Categories.Skip(2)
3         select c;
4
5 foreach (var item in q)
6 {
7     // ..
8 }
```