Building Web Application with ASP.NET MVC using Azure DocumentDB

> DocumentDB 是 Microsoft 提供的 Document-Based NoSQL方案，具備NoSQL資料庫的優點，但同時可以使用SQL語法查詢資料，對於RDBMS 的使用者可以無痛學會如何查詢資料，微軟的OneNote就是使用DocumentDB來提供服務。

使用Azure Portal 建立 Azure DocumentDB

1. 建立DocumentDB帳戶(產生URI&金鑰)
2. 建立DocumentDB資料庫
3. 建立Collection(集合)
4. 建立文件

建立ASP.NET MVC存取 Azure DocumentDB

1. 建立空白ASP.NET MVC 專案
2. NuGet安裝 DocumentDB .NET SDK

```
Install-Package Microsoft.Azure.DocumentDB
```

3. web.config -> 區段加入以下設定

```
<add key="uri" value="your documentdb url" />
<add key="authkey" value="your key" />
<add key="databaseid" value="your databaseid" />
```

4. 建立Model -> ToDoItem : DTO類別

```
public class ToDoItem
{
    public String EventId { get; set; }
    public String Title { get; set; }
    public String Content { get; set; }
}
```

5. 建立Model -> ToDoItemService : Connect Azure DocumentDB Service類別

```
public class ToDoItemService
{
    private static string _dbid;
    private static string _uri;
    private static string _authkey;

    public ToDoItemService()
    {
        _dbid = ConfigurationManager.AppSettings["databaseid"];
        _uri = ConfigurationManager.AppSettings["uri"];
        _authkey = ConfigurationManager.AppSettings["authkey"];
    }

    public List<ToDoItem> QueryAll()
    {
        List<ToDoItem> result = new List<ToDoItem>();

        try
        {
            using (DocumentClient client = new DocumentClient(new Uri(_uri), _authkey))
            {
                var db = client.CreateDatabaseQuery()
                            .Where(d => d.Id == _dbid)
                            .AsEnumerable()
                            .FirstOrDefault();

                var collection = client.CreateDocumentCollectionQuery(db.SelfLink)
                    .Where(c => c.Id == "item")
                    .AsEnumerable()
                    .FirstOrDefault();

                foreach (var item in client.CreateDocumentQuery<ToDoItem>(collection.SelfLink
```

```
                , "select * from item "))
            {
                result.Add(item);
            }
        }
    }
    catch (Exception)
    {
        throw;
    }
    return result;
}

public void Add(ToDoItem item)
{
    try
    {
        AddItemAsync(item);
    }
    catch (Exception)
    {
        throw;
    }
}

private static async Task AddItemAsync(ToDoItem item)
{
    using (DocumentClient client = new DocumentClient(new Uri(_uri), _authkey))
    {
        var db = client.CreateDatabaseQuery()
                    .Where(d => d.Id == _dbid)
                    .AsEnumerable()
                    .FirstOrDefault();

        var collection = client.CreateDocumentCollectionQuery(db.SelfLink)
            .Where(c => c.Id == "item")
            .AsEnumerable()
            .FirstOrDefault();

        await client.CreateDocumentAsync(collection.SelfLink, item);
    }
}
}
```

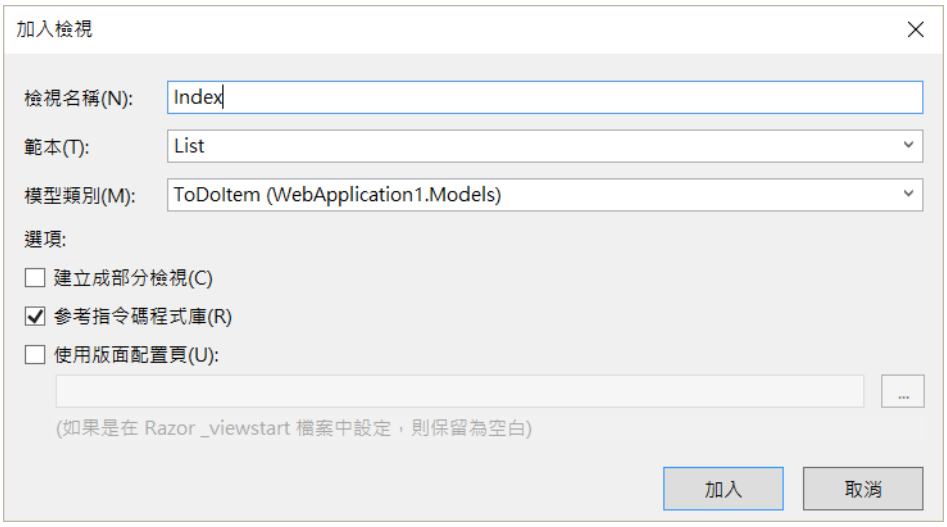
6. 建立Controller ->ToDoItemController

```
public class ToDoItemController : Controller
{
    // GET: ToDoItem
    public ActionResult Index()
    {
        ToDoItemService service = new ToDoItemService();
        return View(service.QueryAll());
    }

    public ActionResult Create()
    {
        return View();
    }

    public ActionResult CreateItem([Bind(Include = "EventId,Title,Content")] ToDoItem item)
    {
        ToDoItemService service = new ToDoItemService();
        service.Add(item);
        return RedirectToAction("Index");
    }
}
```

7. 建立View -> index.cshtml



透過Visual Studio 2015 直接連接Azure DocumentDB