ArchPack Training Templates

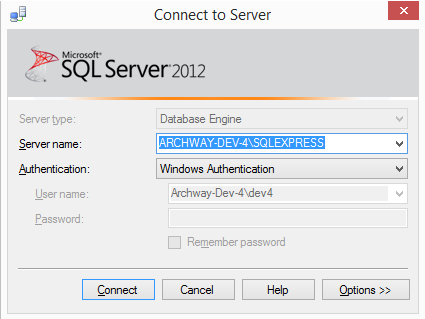
**Set up database (use SQL Server 2012)**

1. Check [Server name] when you login in your SQL Server

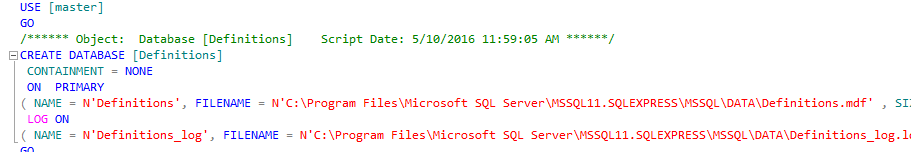
Server name: {your name computer}\ {SQLEXPRESS}

Authentication: No account

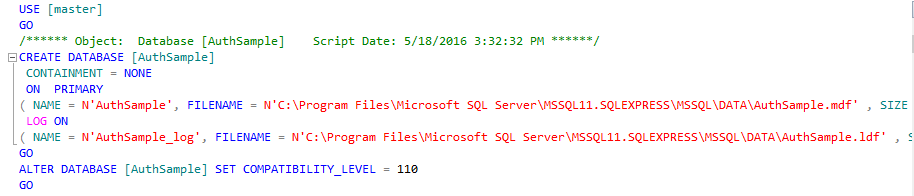
* + - You can follow execute query to create new database in below



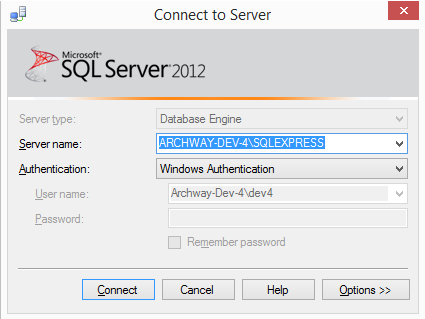
* + - Execute [definition.sql] (Arckpack.Training/Database/definition.sql) to create database



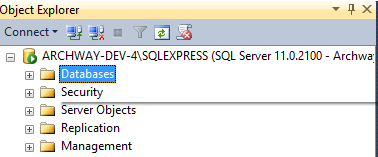
* + - Execute [AuthSample.sql] (Arckpack.Training/Database/AuthSample.sql) to create database



1. If your [Server name] not like format, you can do that to execute query to create new database
   * + - Click [Connect] \*sure the connection is OK\*



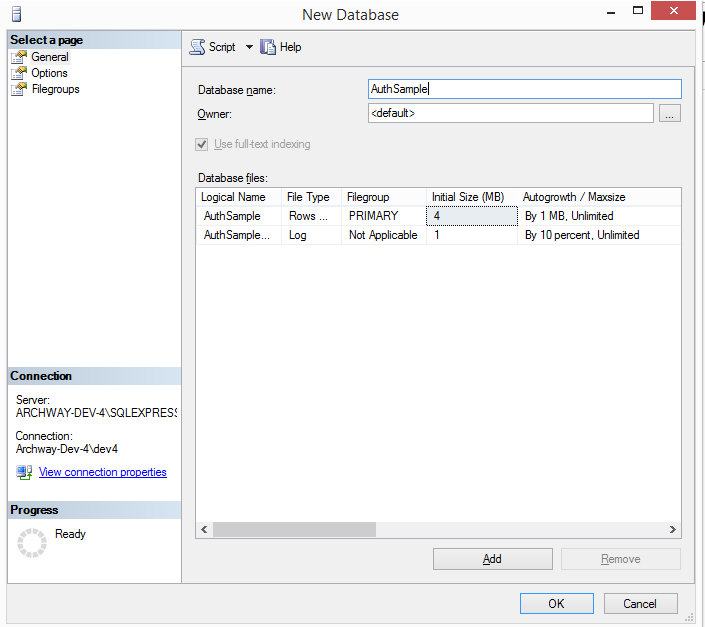
* + - * After login, Right click [Database] 🡪



* + - * Choose [New database]

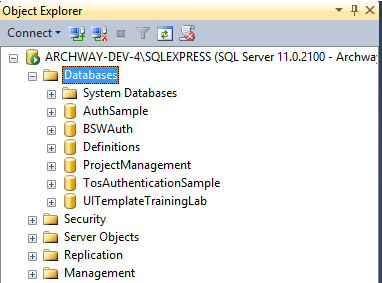
Database name: [AuthSample] have to the same name \*Check query in AuthSample.sql\* like that 

* + - * Click [Add]

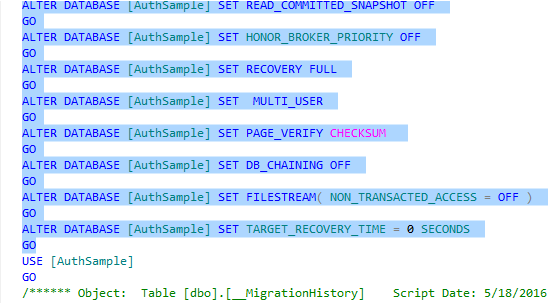


To create new database [Definitions] is the same way.

1. After you create two database name: [AuthSample], [Definitions]



* + - Right click database name [AuthSample] 🡪 [New query], Copy query content [AuthSample.sql] and delete …

* + - Execute from: Use [AuthSample] Go to the end.

To create new database [Definitions] is the same way.

**Set up Templates**

* + - These are two templates, (Archpack.Training/Templates) folder

[UI: {.aspx}, {.datacontext.js}, {.js}, {.less}]]

[API Controller: {controller}, {SearchController}, {CountController}]

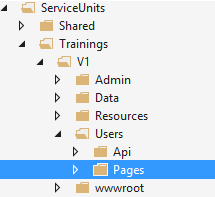
* + - Unzip 2 templates, Copy 2 templates into ->

My Document -> Visual Studio 2013 -> Templates -> ItemTemplates -> Visual C# -> Web ->

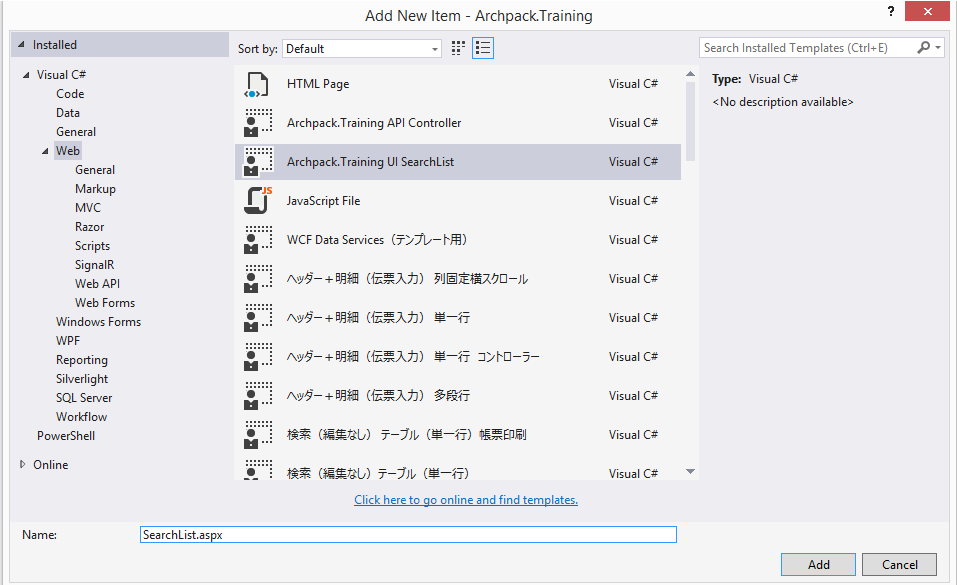
* + - Open visual studio -> File -> Open solution project (.sln) (Direct to path contain project)

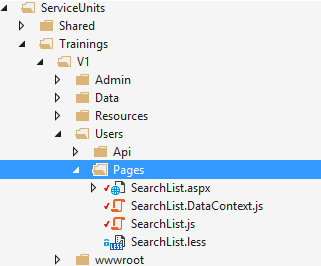
\*If not see template when add new item in your project, please delete this folder: C:\Users\{nameComputer}\AppData(hidden folder)\Roaming\Microsoft\VisualStudio\12.0\ItemTemplatesCache \*

1. ***Add UI Templates***
   * + - Click tag Solution Explorer, **right** click [Pages] folder -> Add new item ->

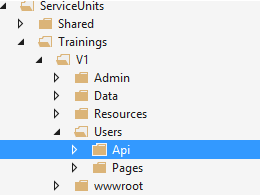


* + - * Edit Name: {name}.aspx

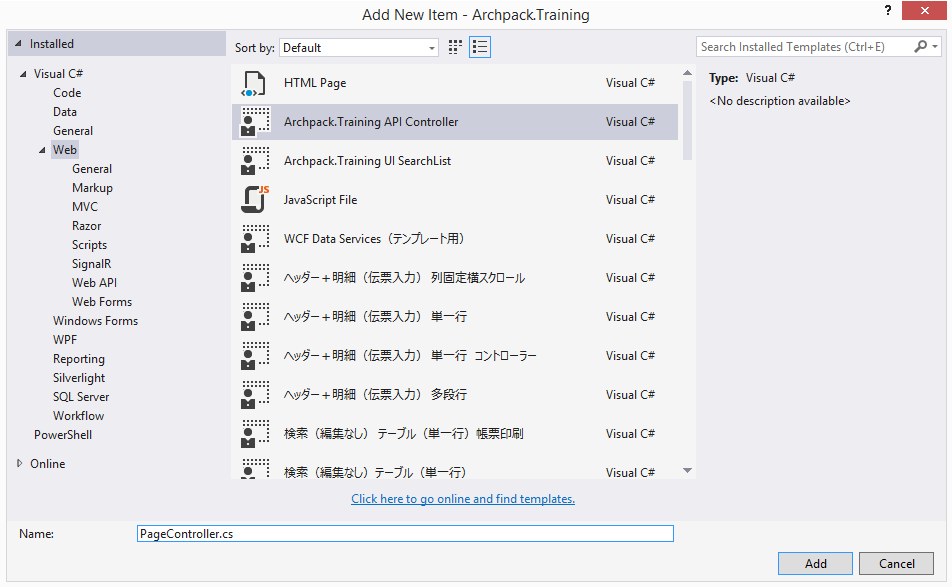


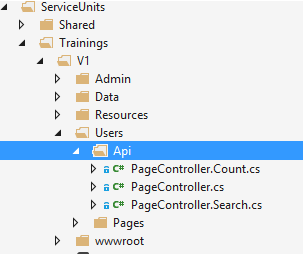


1. ***Add controller templates***
   * + - Click tag Solution Explorer, **right** click [Api] folder -> Add new item ->



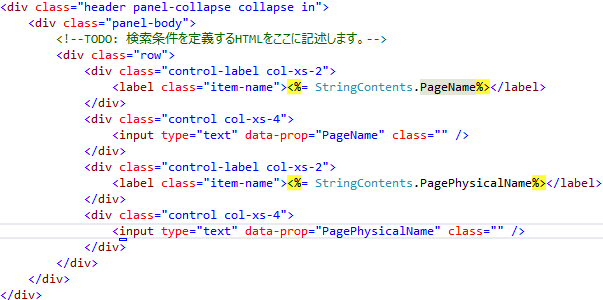
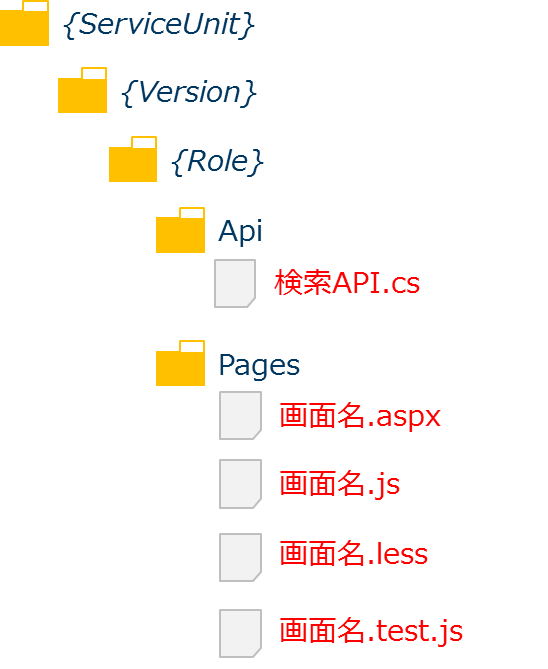
* + - * Edit name: {name}{Controller}.cs





1. **UI SearchList**
   * + *ASPX*

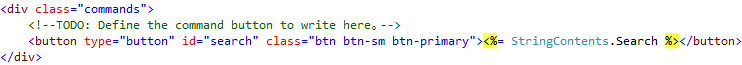
Header content: describle HTML to define header in search list. Base on field in table[Page] get two field (Name, PhysicalName)



* + - * + Detail content



* + - * + Define command button search



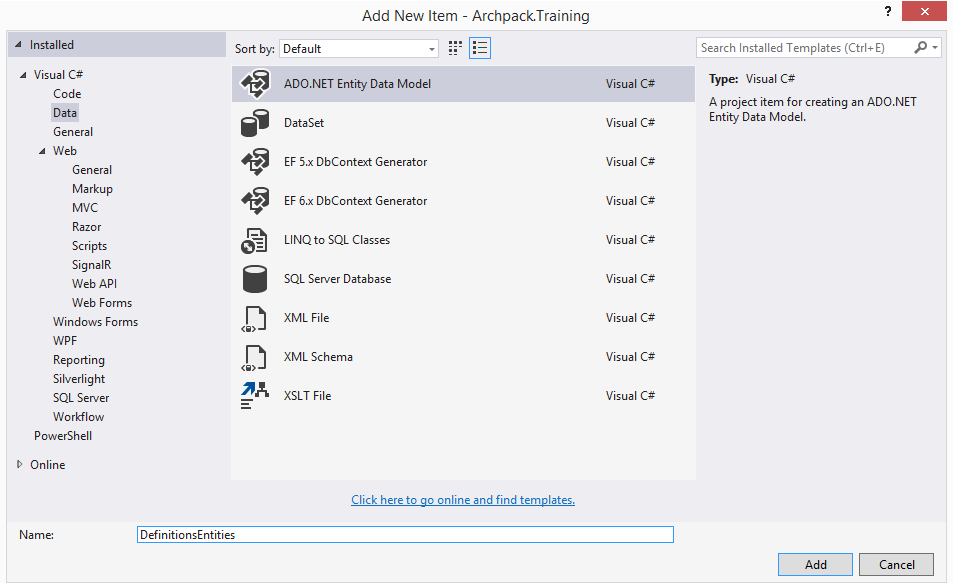
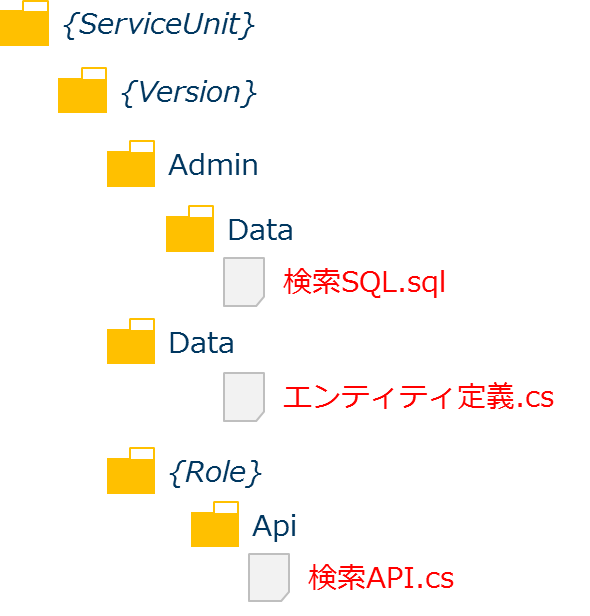
* + - * + Add the path of the script



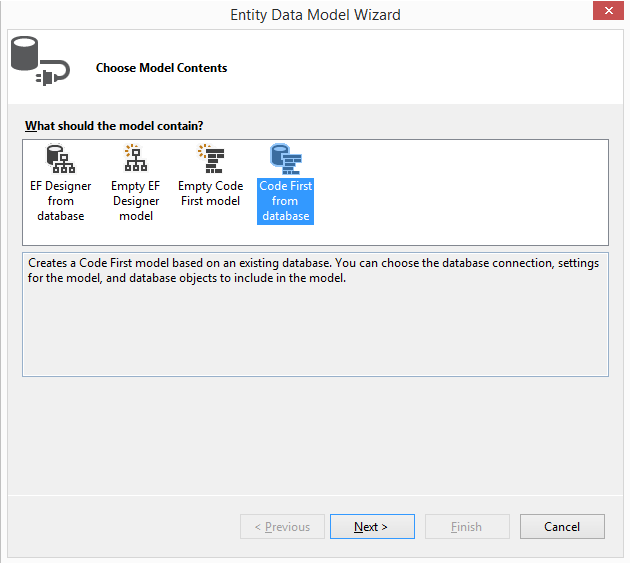
1. **SERVICES**
   * + ***Data Entities***
       - * Create new class generate from Entities framework

V1 / Data folder: right click Data folder, add new item, choose Data 🡪 ADO.Net Entity Data Model

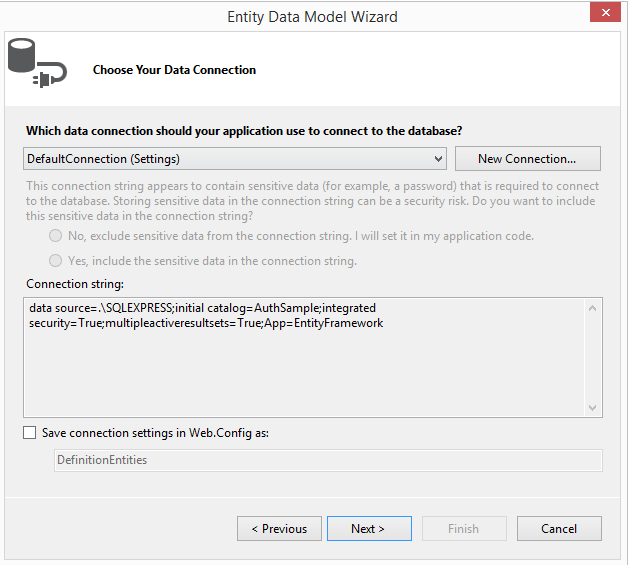
Name: DefinitionsEntities 🡪 Click Add

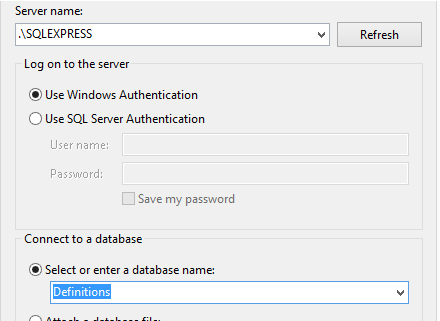
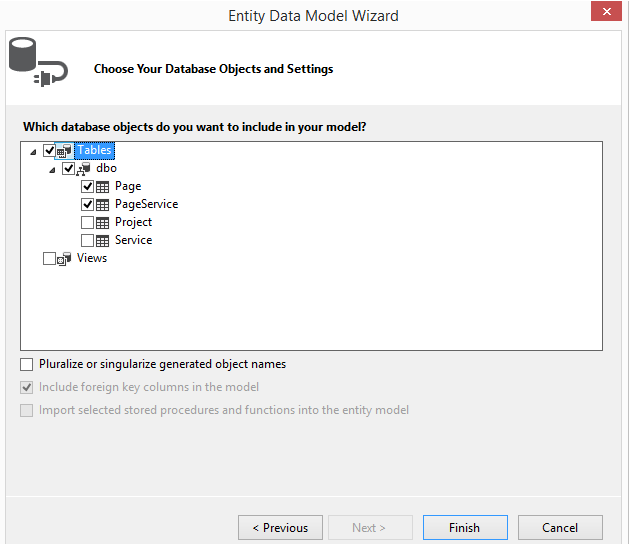


* + - * + Choose [Code First from database] \*database exist\*



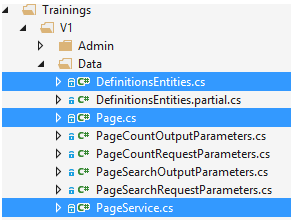
* + - * + Click [New Connection] to create new database \*Uncheck “Save connection…”\*

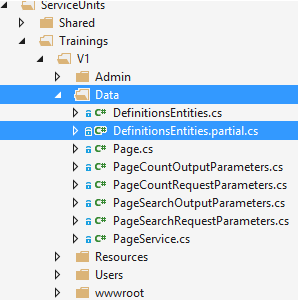




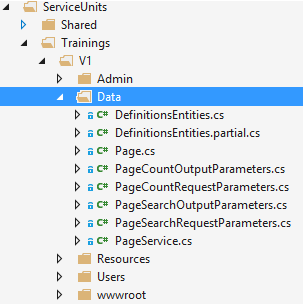
* + - * + After you created new connection, choose table [Tick: Page, PageService] to generate entities class

\*Uncheck “Pluralize or…”\*

* + - * + Generation Entities 2 table: Page, PageService

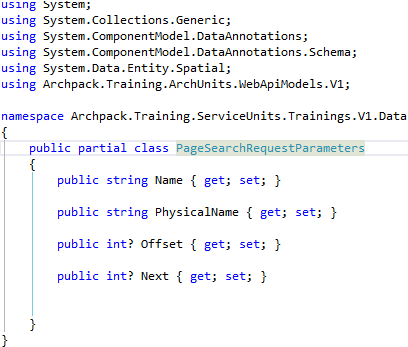
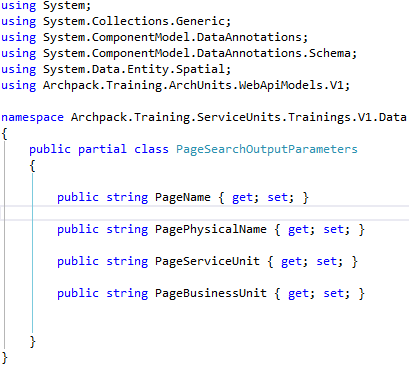


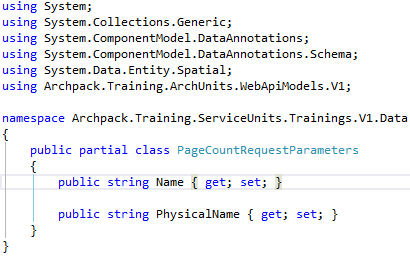
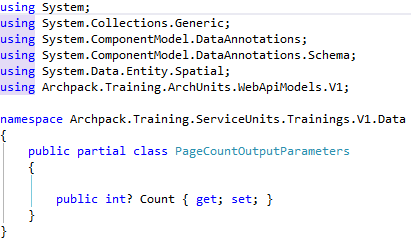
* + - * + Create new class name [ {Entities class}.partial.cs ] to load service config



* + - * + Create new class name: Request vs Output parameters

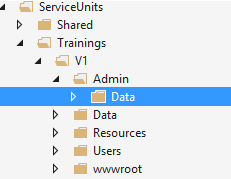
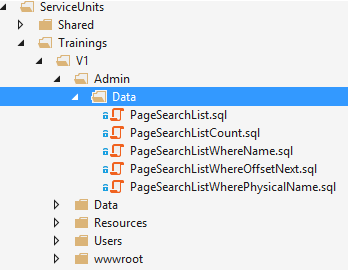
Data folder right Click - Add new item class. Include (SearchRequestParameters.cs, SearchOutputParameters.cs; CountRequestParameters.cs, CountOutputParameters.cs)

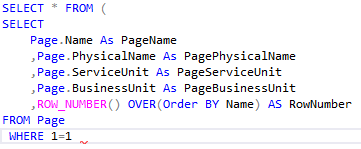
 

* + - ***Sqls***

In Admin/Data, Add – New Item - Choose [Scripts] – Edit name {Name}.sql

* + - * + PageSearchList.sql ( search all data)



* + - * + PageSearchListCount.sql (search all with count data)



* + - * + PageSearchListWhereName.sql (Search with condition name)



* + - * + PageSearchListWherePhysicalName.sql (Search with condition Physical name)



* + - * + PageSearchListWhereOffsetNext.sql (Search with condition nextSearch)

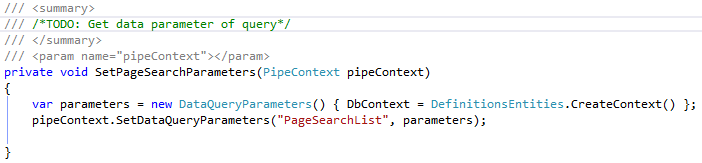


* + - ***API***
      * + PageController.cs



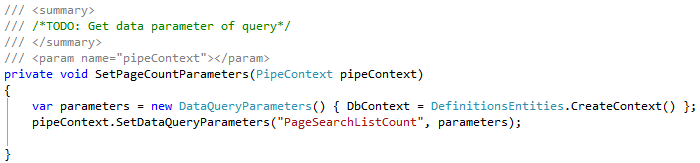
* + - * + PageController with [Search] method



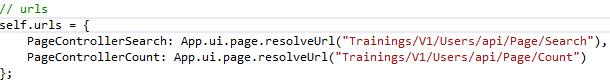


* + - * + PageController with [Count] method

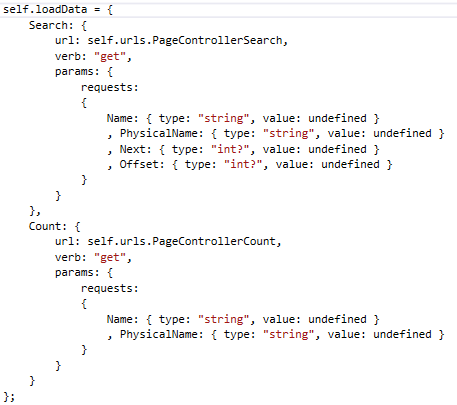




* + - ***DataContext.js***
      * + Set api controller urls to get service



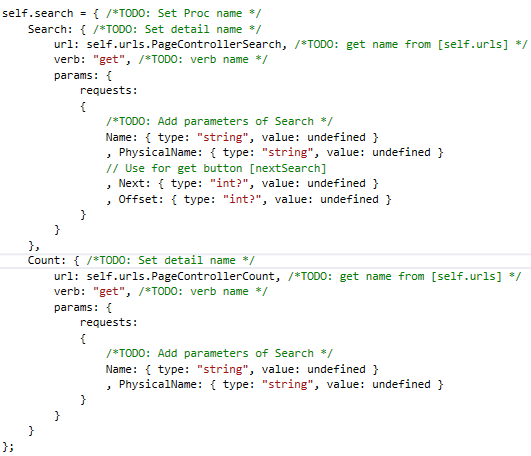
* + - * + Set proc [loadData] to load all data when run the first times



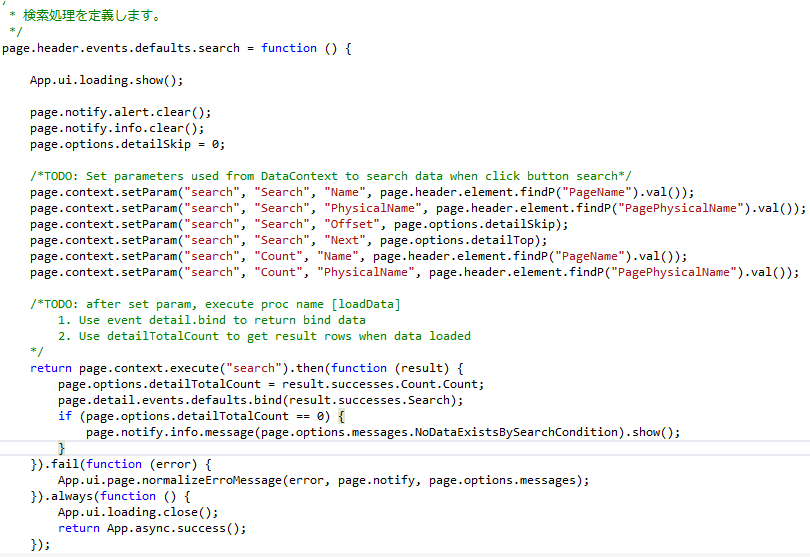
*JavaScript.js file*



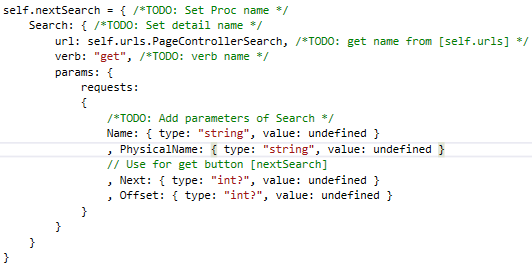
* + - * + Set proc [search] to use for button search when Click



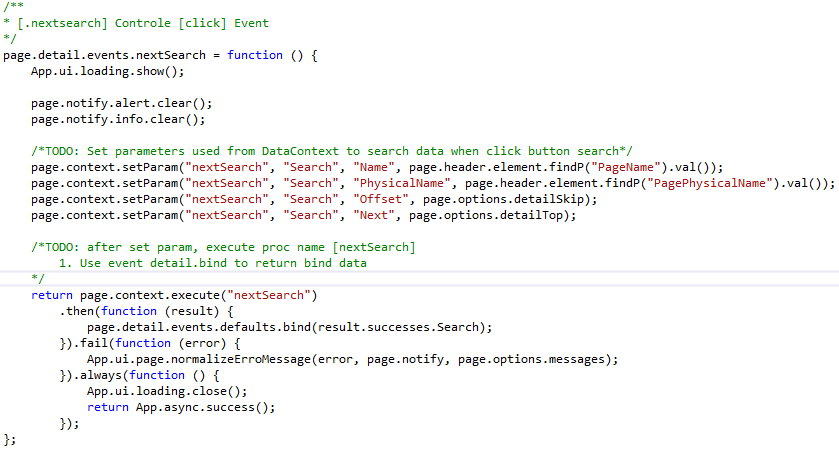
*Javascript*.js *file*



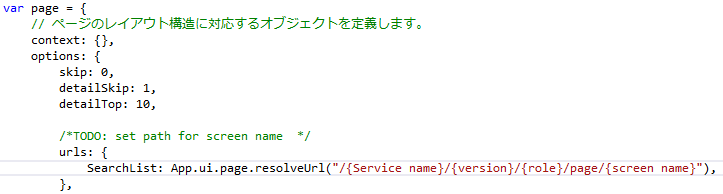
* + - * + Create event next search



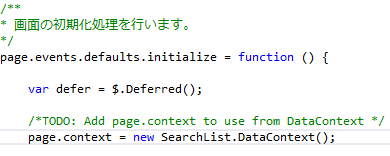
*Javascript*.js *file*



* + - ***JavaScript***



* + - * + Call file DataContext.js



* + - * + Create event click for button search



* + - * + Add button next search

