VEVO INTERFIEW TEST APPLICATION FOR FESTUS WANJOHI

# Overview

The project consists of the following components :

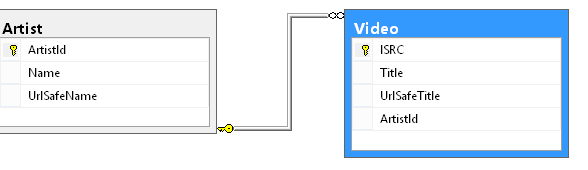
* Database
* Data Access Repository Service
* RESTFul MVC WebAPI End Points
* UnitTests

**Database**

* Used SQL Server 2014, since that’s what I had on my laptop, but it should work on Sql Server 2012 too.
* I had problems creating the DTS Package to import the data provided in the instructions, so I created a simple SQL Server script to import the data into the appropriate tables.
* Also for some reason, I had issues generating the final Database Scripts for creation of a new database from scratch, using the MSSQServer Management studio. I have included the scripts, just in case its my SQL Server environment messing up
* As a contingency measure, I created a backup from my database, just in case you have the same problems as I did. If this step is needed, all you need is to restore the backup and everything should work
* DataBase name use is VEVONEW, but that can be changed in the Web.Config file.

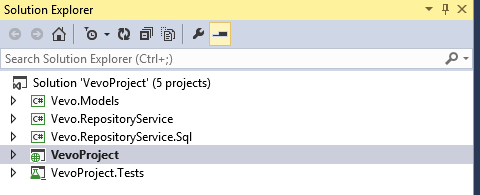
**Why this schema?**

* I decided to use the hybrid between Natural Keys and Surrogate Key. Based on the data provided.
* For the Video table, I realized that ISRC keys are pretty short and looks like they are coded to some standards, thus it might not be too hard to use that in queries.
* For the artist table, the I added a Surrogate key. Irealized that the updateArtist API could change the urlSafeName natural key. This would mean that since there is FK relationship video and Artist, you would have to update the new change in the Video table which could be tasking.
* I added indexes on the serchable natural keys to speed up looku using UrlSafename.



The **Solution**

The back end consists of the following projects



VevoProject -This is the RESTful API End Points

Vevo.Models – This is the POCO representation of objects used throughout the applications, regardless of what DataAccess Service you are using This layer has no logic whatsever, and is thus referenced throught out the project.

Vevo.RepositoryService – This Project contains the Interface contract that is used for any data access service that can be used by the API. It also contains a Factory that reads the web.config to determine what instance of a DataAccess Service to use.

Vevo.RepositoryService.Sql – This is my personal implementation of the IRepositoryService, which accesses data from the SQL Server Database, Does all the business logic needed to process the data and present it to the back end.

This Instance can be interchanged by simply changing the Web.Config to use any instance that implements the IRepositoryService interface.

* *NOTE : This project should not be referenced anywhere in the project, since it will be instanciated by the RepositoryServiceFactory at run time following Dependency Injection Pattern.*
* *HOWEVER : One has to be careful during configuration of the project in that – in my code the dll has to reside in the /bin folder of the deployed project.*
* *THUS: In this solution, I have referenced it in the VevoProject, to make it run out-of-the box, since the file will be automatically copied to the bin folder. This is not a necessary step if you remember to copy it to the bin folder during your deployment, or before running the Project otherwise all the tests will fail, and you will get this message :*
* {
* "Message": "service Provider not well configured : File C:\\git\\Vevo\\Web\\VevoProject\\VevoProject\\bin\\Vevo.RepositoryService.Sql.dll not found"
* }
* *The file is located here : VevoProject\Vevo.RepositoryService.Sql\bin*

VevoProjects.Tests – These are the unit test used to make sure that the functionality works.

# STEPS TO MAKE IT WORK.

* Get the Database up and running – this can be done either from the script folder \DBAssets\VEVO.sql , or by restoring the BACKUP located in \DBAssets\vevotest.bak
* You can edit the script to name the database in case it conflicts with others.
* Go to the Web.Config file and make sure the following information exists : Make sure the RepositoryData points to the right connection string

<!--type data for the repository to use;-->

<add key ="RepositoryProvider" value ="Vevo.RepositoryService.Sql.dll,Vevo.RepositoryService.SqlService"/>

<!--Any configuration data needed by the repository-->

* <add key = "RepositoryData" value="Server=(local);Initial Catalog=VevoNew;Integrated Security=SSPI"/>
* Compile the code to make sure all the items are installed.
* Run the code to make sure that the project runs.
* Make a quick test by typing the URL <http://localhost:38311/api/artists/>
* If all is installed properly, and the port is correct, you should see some data returned.
* Run the Unit Tests.

**Ponts of Failure**

* Database scripts not installed properly,
* IIS not running
* The file Vevo.RepositoryService.Sql.dll not in the /bin folder of the main project
* Web.Config not well set
* Any other Code Issues. LETS DEBUG !!!! ☺

**Conclusion**.

Was fun working on this project. Hope to hear from VEVO, and I hope to be working for the company soon.