**Hospital Simulator Project**

Danielle Hu

# Environment:

Visual Studio Professional 2015 Version 14.0.2543.01 Update 3

Windows 7

# System Design:

This project is a Restful Asp.Net Web API (5.2.3) program. It is designed using a metadata driven and loosely coupled layer architecture. Here is the layer structure:

**DataAccess Services 🡪 Business Services 🡪 HospitalAPI (WebAPI)**

The dependency Injection and resolver between layers are handled using the Unity Container and MEF (Managed Extensibility Framework).

# To make the system flexible and extensible, the business logics that may vary from hospital to hospital or change frequently are implemented in the data model as metadata. One example is the relationships between doctor roles and patient conditions. The original data model is detailed in the file “Data\_model.jpg”. The implementation of this data model in the DataAccessLayer is simplified slightly.

# Code Structure – Five projects

* DataAccessLayer – data access services repository.
* BusinessLayer – business services
* HospitalAPI – WebAPI
* HospitalAPI.Tests – unit tests for HospitalAPI (only few test cases implemented)
* DependencyResolver – handle the dependency Injection and resolver between layers

# Hosting

It is hosted in a local IIS with a website named “HospitalAPI” and a physical location “C:\HospitalAPI”

# API Documentation

File name: Hospital\_Web\_API\_Help\_Page.htm or <http://localhost/HospitalAPI/help> after you publish the API to the local IIS.

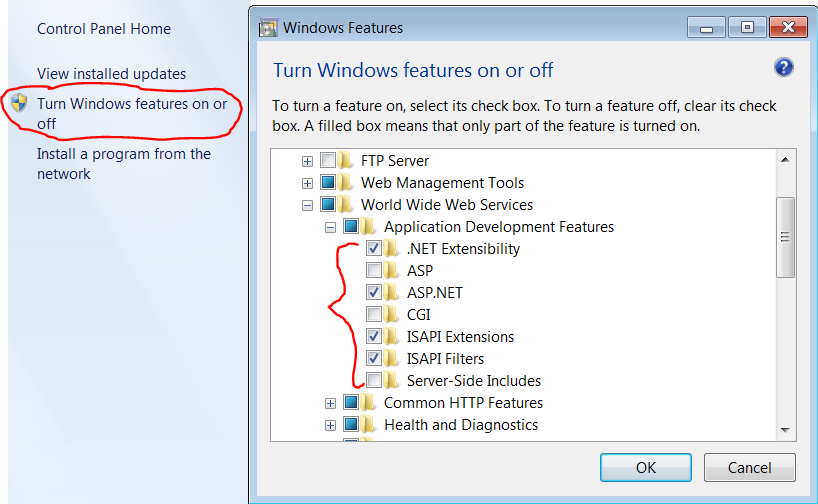
# Others:

* Logs and exceptions – handled globally using NLog. The log directory is HospitalAPI/logs
* Test client – WebApiTestClient by yaohuang
* In-memory Cache – it is not necessary for the project. However, as an exercise, it is implemented in the DoctorsController and the TreatmentRoomsController
* Documentation – Microsoft.AspNet.WebApi.HelpPage (see Hospital\_Web\_API\_Help\_Page.htm for a saved copy)
* Versioning – each controller has a version number as v1, e.g. api/v1/patients
* BusinessLayer unit tests – not implemented
* Concurrency – not handled
* authenticity and authority – not handled

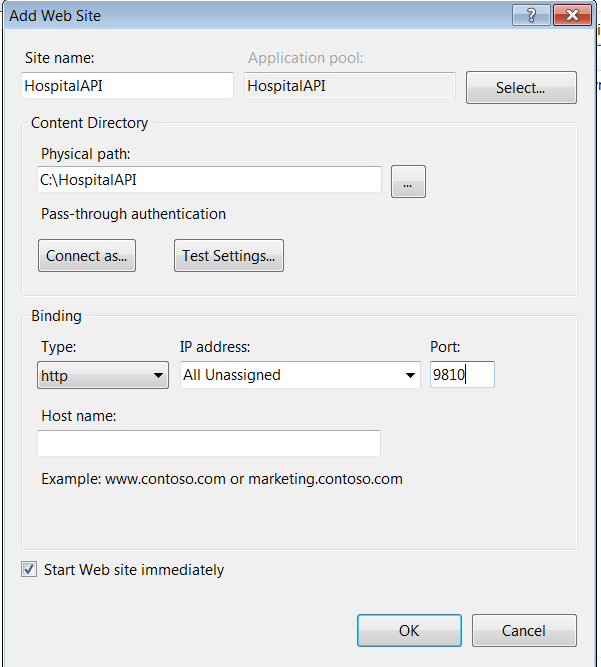
# Setup

1. Install IIS if not there
2. Click on Control Panel -> Programs and Features ->Turn Window features on or off

Make sure .NET Extensibility and ASP.NET are checked



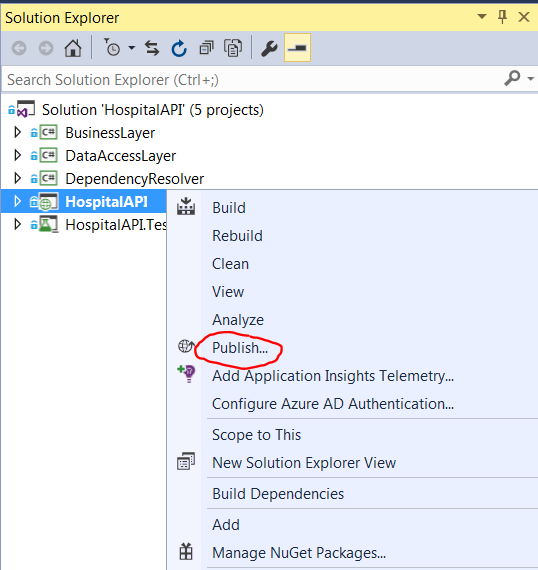
1. Open the Internet Information Services (IIS) Manager in Control Panel -> Administrative Tools
2. Create a site named “HospitalAPI”. I set physical path as “C:\HospitalAPI” and port 9810. You can use any physical path, just make sure IIS has permission to this directory.



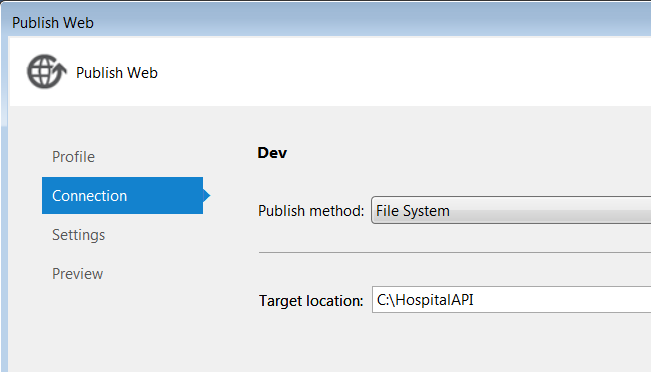
5. Download the source code from this github: <https://github.com/Griffin8/HospitalAPI.git>

6. Open the HospitalAPI.sln using Microsoft Visual Studio 2015 and compile the solution.

7. Right click on the project HospitalAPI and select Publish to publish the web API to the local IIS site

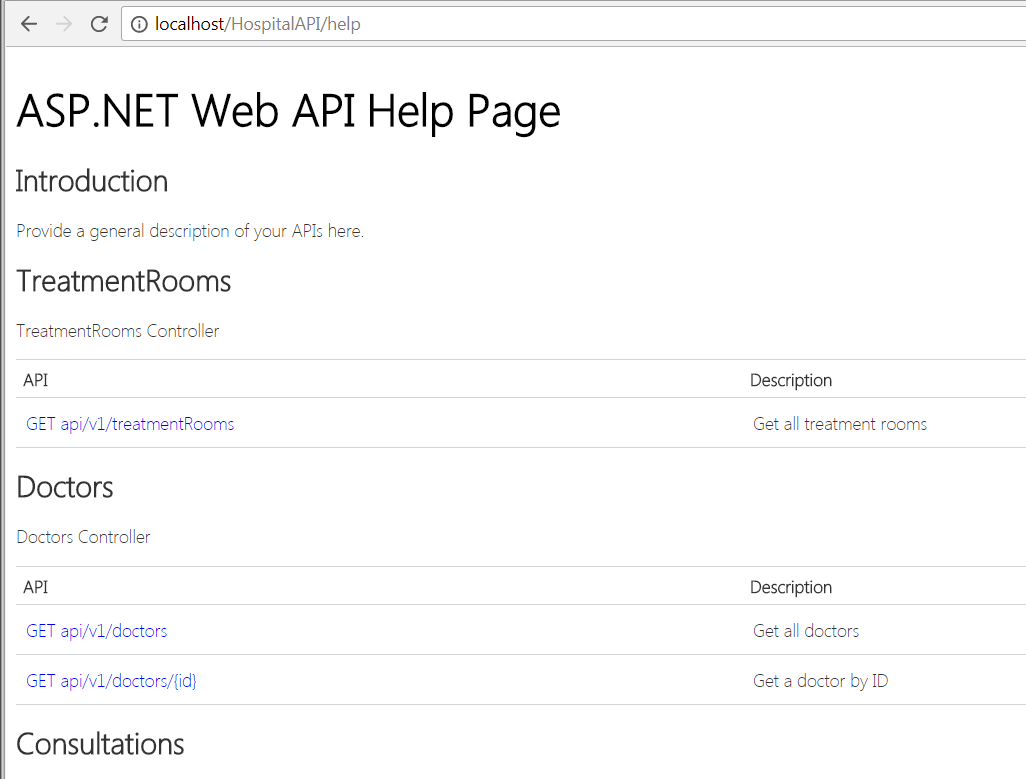


8. The Profile “Dev” is configured with a target directory “C:\HospitalAPI”. You should change it to match the directory you choose in the step 4



9. Go to your browser and enter <http://localhost/HospitalAPI/help>

You should see a page like this



10. Click on any link in this page, you will see a button “Test API” at the lower right corner. Click on this button and you can test the API. Enjoy!

