

VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY  
INTERNATIONAL UNIVERSITY  
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING



**GUIDELINE FOR THESIS' PROJECTS SOURCE CODE AND  
DEMONSTRATION**

by  
Pham Le Trung

A guideline submitted to the School of Computer Science and Engineering in  
partial fulfilment of the requirements for the degree of  
Bachelor of Computer Science

Ho Chi Minh City, Vietnam  
2018

## TABLE OF CONTENTS

INTRODUCTION .....	3
CHAPTER 1 NETWORK CONFIGURATION.....	4
1.1    SYSTEM'S CONNECTIVITY.....	4
CHAPTER 2 SERVER SETUP .....	6
2.1    THE DATABASE .....	6
2.2    Web API.....	7
2.2.1    Source Code .....	7
2.2.2    Publish the Project .....	7
2.2.3    Hosting the Server WEB API Using IIS .....	8
CHAPTER 3 MOBILE - CLIENT SETUP .....	12
3.1    Server Request Configuration.....	12
3.2    Installing the Code .....	13

## INTRODUCTION

This document is served as the sub-component for the original thesis topic “Development of Mobile Peer-to-Peer Application for Customer-to-Customer E-Commerce”. The main purpose of the guideline is to show how to configure the system from one element to another in the setup order.

It is recommended that in order to have full understanding of how source code is organized, the structured and the reason behinds the design, please kindly read through the thesis at Chapter 3 - Method and Implementation.

If after following both the original thesis and this guideline document, still you cannot set up the system, feel free to contact me at [trunglepham1202@gmail.com](mailto:trunglepham1202@gmail.com).

# CHAPTER 1

## NETWORK CONFIGURATION

### 1.1 SYSTEM'S CONNECTIVITY

*All devices* in the demonstration should be run under a LAN, therefore they can communicate through each other easily under the same roof. Different network means the application must handle distinction between public IP and private IP, which is not applicable in the demonstration. Therefore, the IP of all the devices should in all uniform form as 192.168.x.x or 10.x.x.x so that the demonstration could be run appropriately.

For example, demonstration's components IP:

- Server: 192.168.0.107.
- First Device (HTC): 192.168.0.109.
- Second Device (Samsung): 192.168.0.100

To view Server IP on Window, open Command Prompt and type *ipconfig* (*ifconfig* in Linux-based machine), the IP is the IPv4 address line of the output as shown in Figure 1.

```
CA Command Prompt
Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 12:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::91d4:47d:5854:6e6f%7
    IPv4 Address. . . . . : 192.168.0.107
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

C:\Users\trung>
```

Figure 1. ipconfig output

# CHAPTER 2

## SERVER SETUP

### 2.1 THE DATABASE

The structure of the database can be referred in Chapter 3, Section 3.3, Designing the Database in the thesis. The server database would consist of only two table, User and Rating. Any instance of database management system is acceptable.

The demonstration uses Microsoft SQL Server (MSSQL) for storing database for easy compatible with Microsoft Environment as shown in Figure 2.

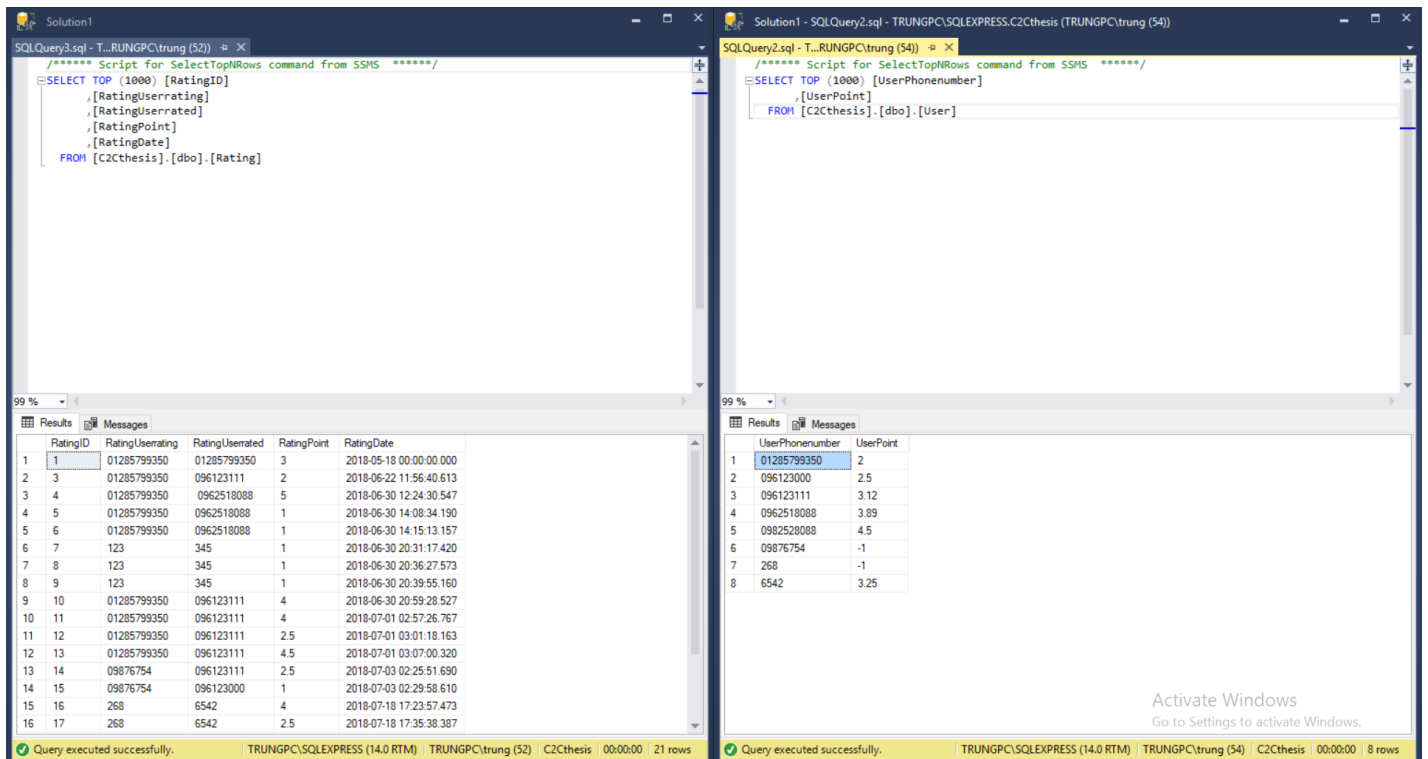


Figure 2. Server Database

## **2.2 Web API**

### **2.2.1 Source Code**

The source code for Web API Server is publicly stored in git repository at [https://github.com/zovippro1996/ThesisProject/tree/master/WebApi\\_C2C](https://github.com/zovippro1996/ThesisProject/tree/master/WebApi_C2C) where you can clone or download to local computer.

The descriptions of important code repositories are mentioned in the thesis at Chapter 3, Section 3.5, Setting up Server Application in the Thesis Document. There are two things to mention in this section:

- Controllers are located under “Controllers” Repository of “Web\_APIC2C” Package. Including UserController and RatingController. In-depth description of how to create Get, Post, Put and Delete for Web Api is reference at Microsoft’s Docs page about Web Api.
- Models are located under package “Web\_APIC2CDataAccess”, inside Web\_APIC2CDataModel.edmx file. Including User.cs and Rating.cs

Other files are just scaffoldings from initialize the Web API and Entity Framework project in Visual Studio 2017. WebApiConfig.cs in App\_Start folder helps setting the URL for Api Calling.

### **2.2.2 Publish the Project**

Publishing the application generates the set of files that are necessary to operate the application. Therefore, IIS could host our application at certain port for other users to send request to our Web Api. Of course, this section is optional if you consider other method of hosting the Web Service.

In order to publish the Application, clicks menu Build/Publish Web, select the Path and press Publish.

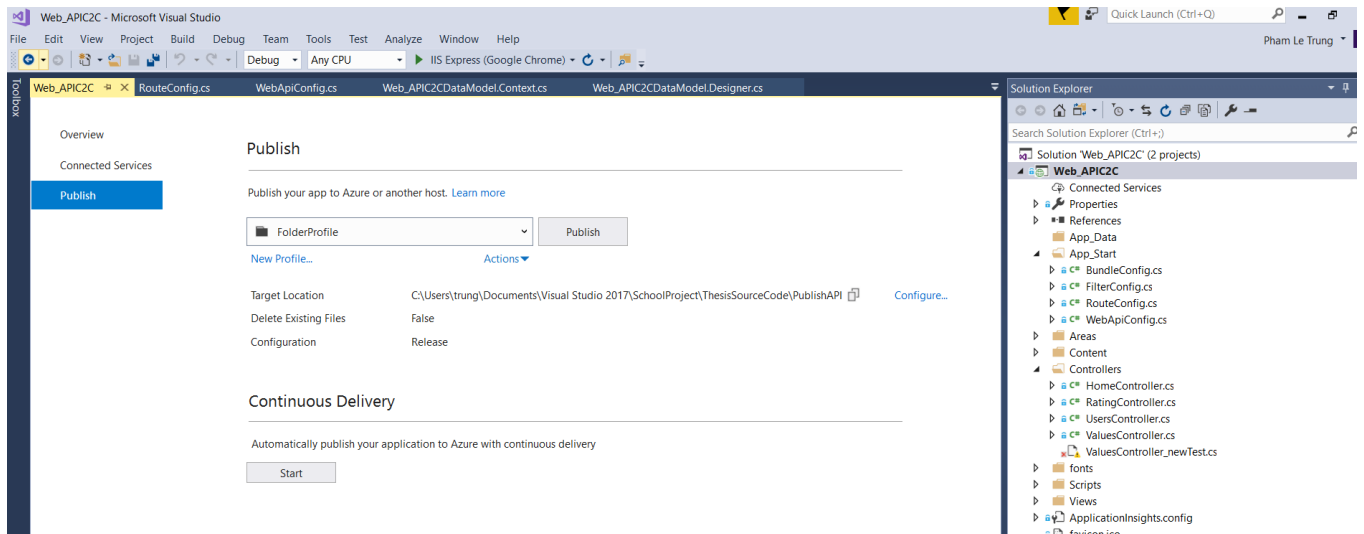


Figure 3. Publish Window

After that, the application generates a publish folder that we can use to host our Web Api.

### 2.2.3 Hosting the Server WEB API Using IIS

The mini Server for the application is run on Window Machine hosting by IIS (Internet Information Server) – extensible web server created by Microsoft, hidden as a feature on any Window Computer. IIS can be activated from the “Window Feature” when searching “Window Feature” on Window Search (Figure 4).



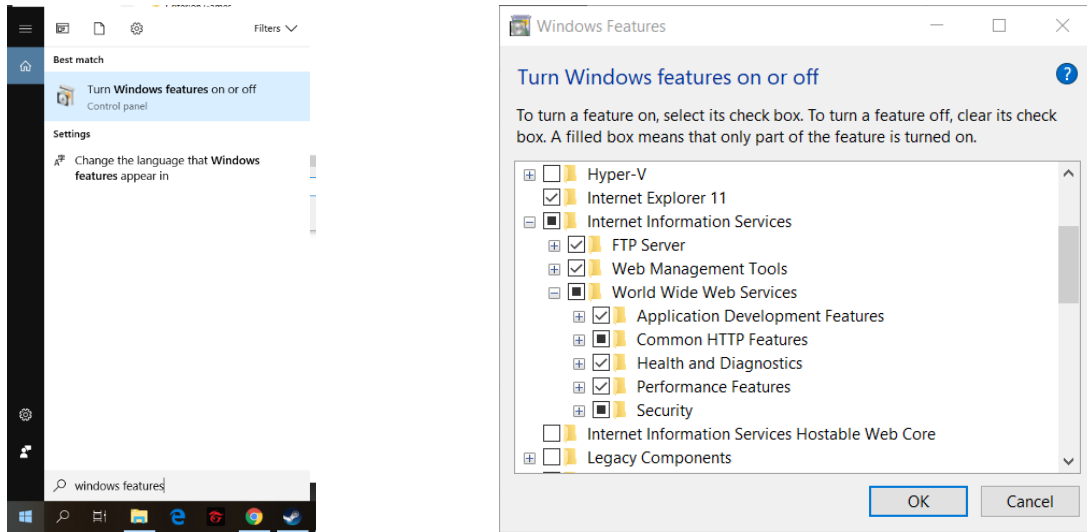


Figure 4. Activate IIS from Window Feature

After activate IIS from Windows Feature, we can open IIS Manager, Add New Website, set the physical path (pointing to our Web Api publish folder), set the port the application.

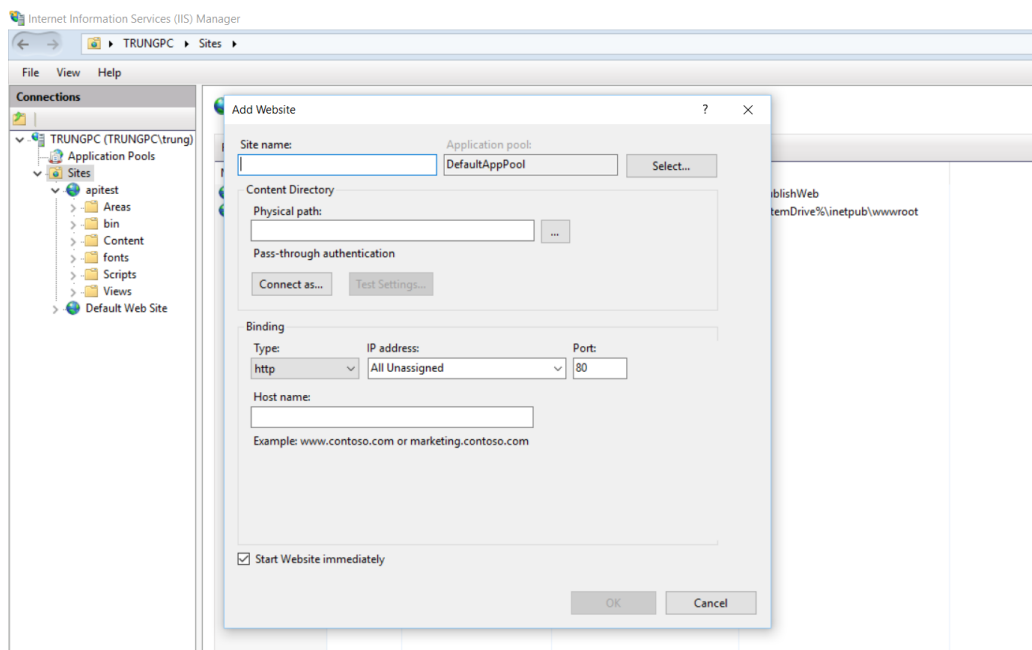


Figure 5. Add Website in IIS Manager

The port is also very important, some machine in default, locked the port that prevents other devices to connect. Additionally, on Window Machine, we should open the hosting port in firewall, so that our client devices could send request to the server.

From window, open “Window Defender Firewall with Advanced Security”, create New Rule for the Inbound Rule. In the New Rule Wizards, select Port, specify the local port you want to open (the one you assign in IIS), then Allow all Connections, assign the name for the rule.

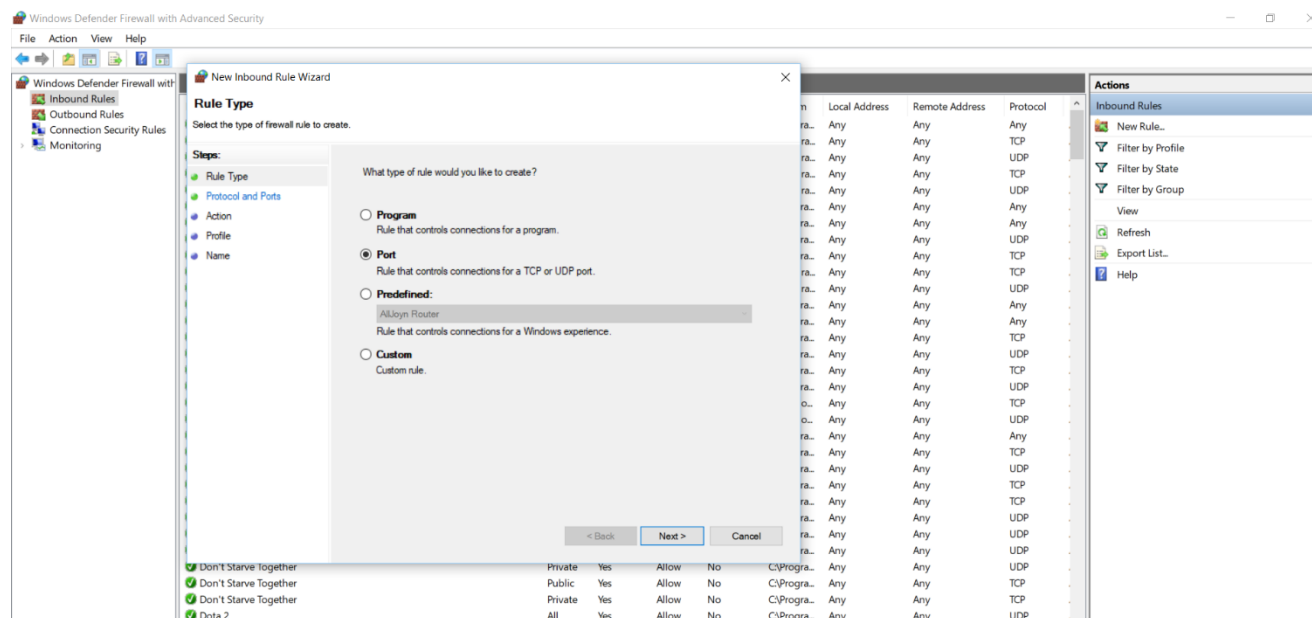


Figure 6. Open Port in Firewall

To test for the web application, simply open web browser, and type in the URL:

*[Your Server IP]:[Port]/*

If there is an ASP.Net Introduction Page as shown in Figure 5, then our Web Api is successfully hosted and waiting for requests from clients.

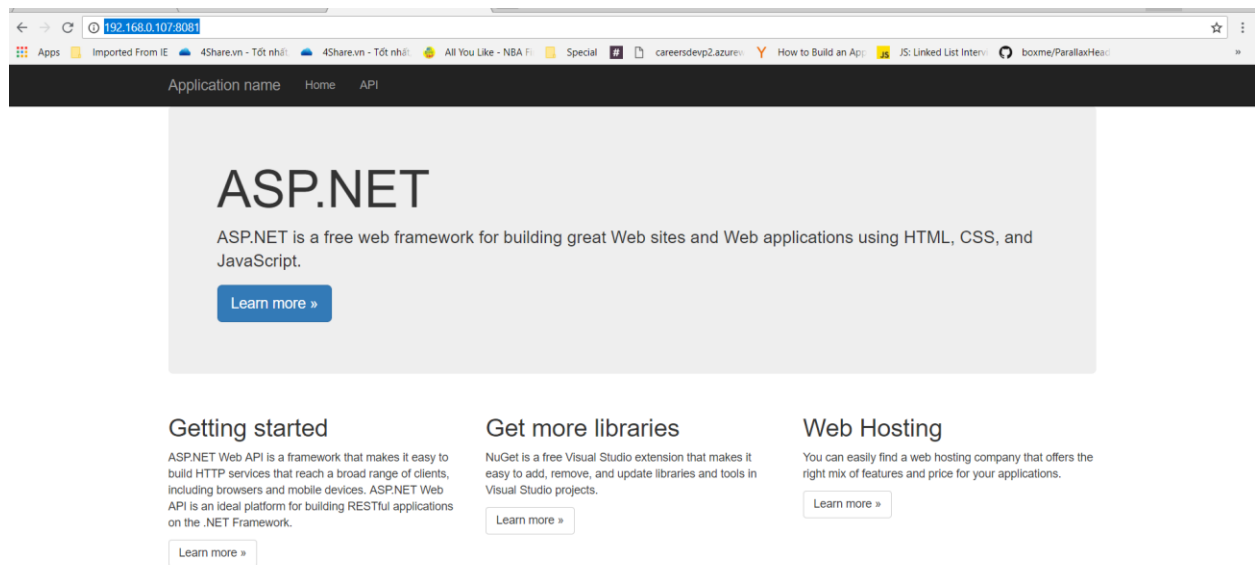


Figure 7. ASP.Net Introduction page

## **CHAPTER 3**

### **MOBILE - CLIENT SETUP**

#### **3.1 Getting the Code**

The source code for mobile application is publicly stored in git repository at <https://github.com/zovippro1996/ThesisProject/tree/master/MobileAppC2C> where you can clone or download directly to local computer.

There are 4 main sections in the source code that we have to know before configuring:

- The gnutellaprotocol package: Storing every functions and object supporting for Gnutella Protocol between devices.
- The helper package: Assist in generating database, creating AsyncTask Socket Connection and generating sub-layout.
- The object package: Help defying and transferring data between activities
- Android Activity Classes: classes that represent screen on Android device, coordinate the flow of application and user interactions.

#### **3.2 Server Request Configuration**

The mobile application has two part that require server requests with URL. If successfully configuring the server as Chapter 2 described, we can easily get the URL of the Web Api that has the form “[ServerIPAddress]:[Port]”.

Simply locate the “TargetProfileActivity” under Java Repository. The URL of Server Requests are defined under “Server URL” Comment. Replace the String with your Server URL. Install the Application to devices and we are good to run the whole systems.

```
//Server URL
String myUrl = "http://192.168.0.107:8081/api/users?phone=" + targetPhone;
//String to place our result in
String result = "";
AsyncTaskIpify getRequest = new AsyncTaskIpify();

try {
    //Server URL
    result = task.execute("http://192.168.0.107:8081/api/rating").get();
    popupToast("Done = " + result.toString());
    Intent intent = new Intent( packageContext: this, DashboardActivity.class);
    startActivity(intent);
}
```

Figure 8. Server URL Declaration in Android application

### 3.3 Installing the Code

After finishing all the setup, the rest is just installing the code to the mobile application either from .apk file or debugging process. Let’s remind that this application is best operates in Android OS later than version 5.0 Lollipop (Figure 8).

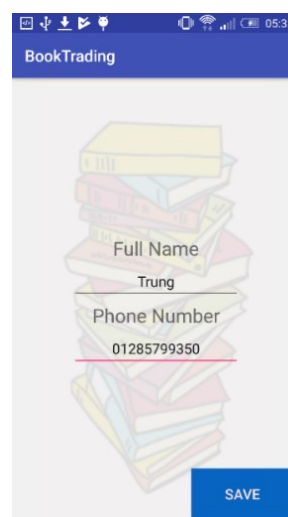


Figure 9. Android Application