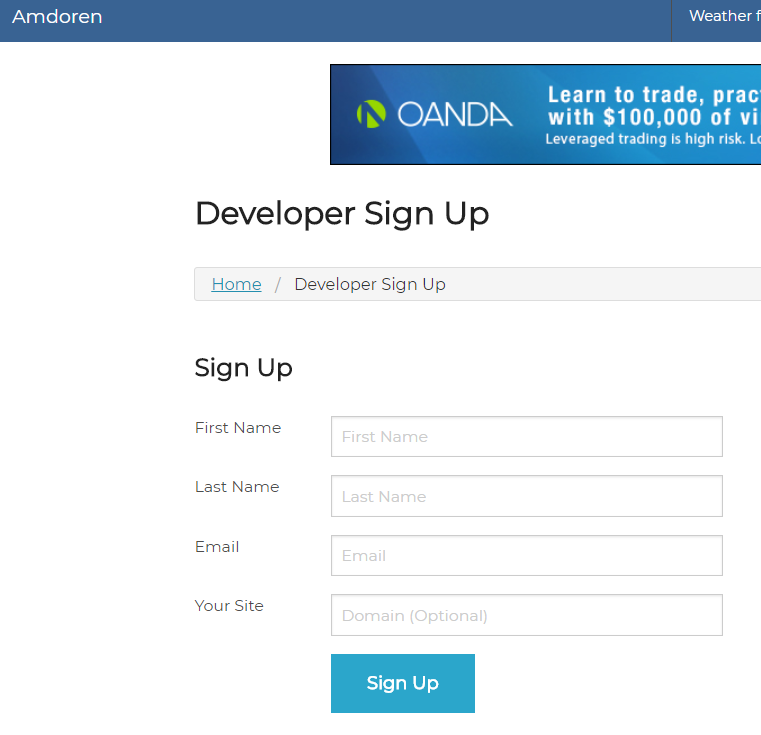
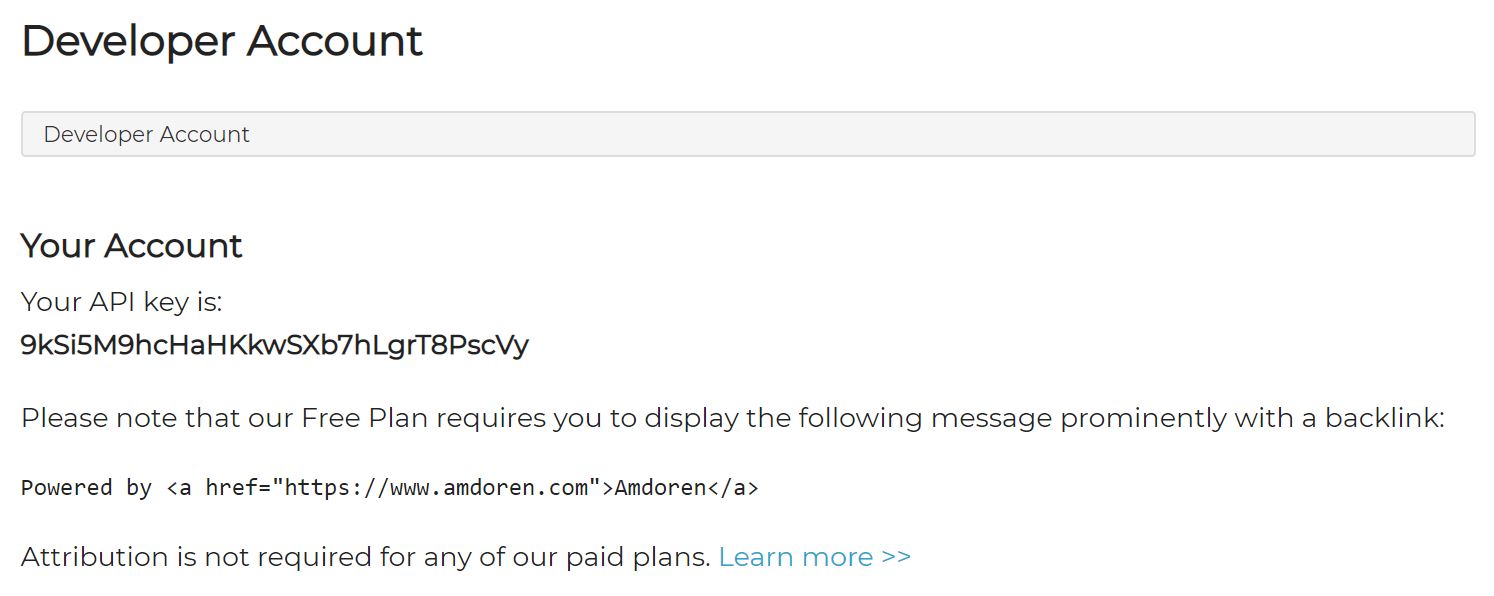
Lesson 11 - Custom Routing & Web API Part 3

1. Setup your project by following the instructions in **C286 L11 Project Setup.docx**.
2. Open **Task List** window from the menu **VIEW**, **Task List**. Use the **Filter** function in the Project heading to display only the Task for **Lesson11**.
3. **Activity 1:**
   * + 1. Sign up for an Admoren developer account at <https://www.amdoren.com/developer/signup/>.

You can use <https://www.minuteinbox.com/> to get a temporary email just for registration purpose in case the API key is expired and you need to register again.



* + - 1. Ensure that you obtained the API Key code:



**Your API key will be different from this one**

* + - 1. Make a call to the Amdoren currency exchange web api using browser (Chrome preferred) or Postman to receive exchange rate from SGD to USD using the following information:

Web API address: https://www.amdoren.com/api/currency.php

Parameters:

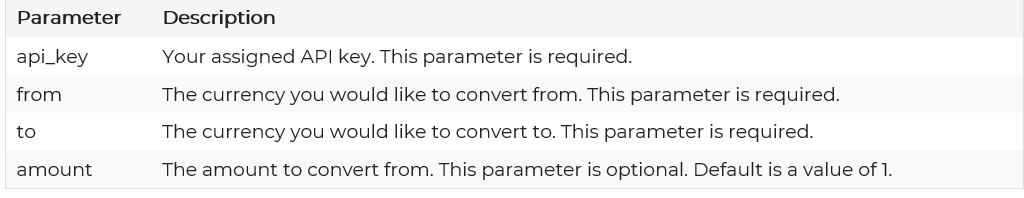
api\_key: **mH48S34y6MVNE7hD9SA2VQtG74MgvN**

from: currency to convert from

to: currency to convert to

1. Write down web API url below:

https://www.amdoren.com/api/currency.php ?api\_key=mH48S34y6MVNE7hD9SA2VQtG74MgvN&from=SGD&to=USD



1. Copy and paste the returned result from the web API in the space below

{

    "error": 0,

    "error\_message": "-",

    "amount": 0.74369

}

1. Write the anonymous type to represent the returned result below:

new

{

    "error": 0,

    "error\_message": "-",

    "amount": 0.74369

}

* + - 1. Call the Amdoren weather web api using browser (Chrome preferred) to receive today and next 4 days weather for Republic Polytechnic:

Web API address: https://www.amdoren.com/api/weather.php

Parameters: https://www.amdoren.com/api/weather.php?api\_key={\_apiKey}&lat={latitude}&lon={longitude}

api\_key: your api key

lat: latitude of the target location

lon: longitude of the target location

1. Write down the web api url below:

**Goto** [**https://www.amdoren.com/latitude-longitude/**](https://www.amdoren.com/latitude-longitude/) **to find out the longitude and latitude of the target location**

Latitude: 1.4428559

Longtitude: 103.7856195

1. Copy and paste the returned result from the web api in the space below

Text

Description automatically generated with medium confidence

1. Write the anonymous type to represent **weather of the 3rd day** below:

new{

“date” : “2022-01-27”,

“avg\_c” : 28,

“min\_c” : 22,

“max\_c” : 35,

“avg\_f” : 82,

“min\_f” : 72,

“max\_f” : 95,

“summary” : “Partly cloudy”,

“icon” : “wi\_color\_partly\_cloudy\_day.png”

}

1. **Activity 2:**
   * + 1. Perform **Tasks 1-1 to 1-5**.
       2. Verify and compare your code with the lecturers’.

1. **Activity 3:**
   1. Examine **Order** controller, fill in the following table based on your analysis of the code witin the class:

|  |  |
| --- | --- |
| Action | Purpose |
| Index | Return string containing all orders detail |
| GetOrder | Return only one order |
| GetOrders | Return multiple orders |
| GetOrdersByDate | Return the orders by date |

* 1. Examine **Product** controller, fill in the following table based on your analysis of the code witin the class:

|  |  |
| --- | --- |
| Action | Purpose |
| Index | Return string containing all products detail |
| GetProduct | Return product |

* 1. Build and run the application, test all actions of **Product**  and **Order** controllers and paste the output into the table for each actions:

|  |  |
| --- | --- |
| Action | Output Image |
| Product/Index |  |
| Product/GetProduct/MO051 |  |
| Order/Index |  |
| Order/GetOrders/01-01-2017 |  |
| Order/GetOrder/0L8013 |  |
| Order/GetOrdersByDate?oyear=2017&omonth=1&&oday=1 |  |

1. **Activity 4:**
   * + 1. Perform **Tasks 2-1 to 2-5**.
       2. Refer to the **Student Notes slides** to help you complete the task.
       3. Test your answers of the exercises using the web request given in the following table and copy and paste the output as well:

|  |  |
| --- | --- |
| Web Request | Output Image |
| Shoppers |  |
| Shoppers/Product |  |
| Shoppers/2017/01/02 |  |
| Shoppers/CZ854 |  |
| Shoppers/2L5481 |  |

* + - 1. Fill in the following table with the corresponding web request in default MVC routing format, {controller}/{action}/{id}:

|  |  |
| --- | --- |
| Web Request | In default MVC routing format |
| Shoppers | Order/Index |
| Shoppers/Product | Product/Index |
| Shoppers/2017/01/02 | Order/GetOrdersByDate?oyear=2017&omonth=01&0day=02 |
| Shoppers/CZ854 | Product/GetProducy/CZ854 |
| Shoppers/2L5481 | Order/GetOrder/2L5481 |

# Solving the Problem

1. You can complete the problem by following the solution tasks in the project. You need to open RPNotes/Index.cshtml to see all solution tasks.
2. Recommended order of implementation:
   1. Custom Routing

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* 1. Weather forecast

-Redid the whole thing because the weather forecast wasn’t working either due to timed out api or incorrect implementation in the code.

Graphical user interface, text, application, email

Description automatically generated

*— End of Worksheet —*