



ICT705

Data and System Integration

Task 2

ATMC Semester 1, 2020

Assessment and Submission Details

Marks: 40% of the Total Assessment for the Course

Due Date: 11:59pm Friday, Week 12 (5th June 2020)

Task 2 consists of a demo code package and a technical report for the integration plan. Submit your work in two parts. Part 1 will include your **report (a .docx file)** and Part 2 is a **zip** file, which contains the **code files (.zip file)**, to Blackboard Task 2. Please follow the submission instructions described in this document.

NOTE: If you do not submit a code file you will receive a 0 for the task.

The task is an **individual assignment** and will be marked out of a total of 100 marks and forms 40% of the total assessment for the course. **ALL** assignments will be checked for plagiarism by SafeAssign system provided by Blackboard automatically.

Refer to your Course Outline or the Course Web Site for a copy of the “Student Misconduct, Plagiarism and Collusion” guidelines. [Academic Integrity Information](#).

Assignment submission extensions will only be made using the official USC Guidelines.

Requests for an extension to an assignment **MUST** be made to the course coordinator prior to the date of submission and requests made on the day of submission or after the submission date will only be considered in exceptional circumstances.

Case Study: Overseas Shipping Logistics Ltd

Overseas Shipping Logistics Ltd (OSL) organises and ships large machinery and boats around the world. It has a similar business model to the following [Link](#).

In your capacity as Information Systems consultant, you have been engaged by OSL to advise them about their global information system architecture. You have advised them about applying a Service Oriented Architecture (SOA) to integrate the dispersed systems in their international offices.

OSL directors would like to gain a deeper understanding of the Technologies behind your recommendations and have asked for a small specific demonstration of these technologies along with a brief explanation of the concepts and principles of how it works and how an SOA could be applied to OSL.

Given the concern over COVID-19 and its impact on the business, OSL directors have asked for a system integration demonstration allowing them to monitor key COVID-19 statistics for particular countries. The demo enables OSL to browser search countries in Google Maps and retrieve current COVID-19 information along with the map.

Assignment tasks

This assignment consists of two deliverables:

1. **Demonstration code** – demonstrating the principles of data merging, RESTful Web Services and Mashups (50%)
2. **Integration report** – explaining the principles of data merging, RESTful Web Services and Mashups and applying them to the case study organisation (50%)

Deliverable 1 – Demo code (50%)

There are three major components in the demo system:

- **Data integration demo.** In this demo, there are two data source files:
 - “owid-covid-data.csv” contains COVID-19 statistical data from over 200 countries for dates ranging from the 31st December 2019 to 30th March 2020*. You may like to download the latest file from [Our World in Data - Coronavirus Source Data](#),
 - “Country_location.xml” contains country location coordinates for each country.

Instructions:

- Before importing the COVID-19 data into Python you are required to:
 - i. Choose 15 countries from the “owid-covid-data.csv” dataset you think would be applicable to the case study,
 - ii. Retain the most current day’s statistics for each of these 15 countries,
 - iii. Add 2 or 3 extra columns to your dataset. These columns will contain other 2 or 3 other interesting statistics *of your choosing* for each of your

* Data is freely available from <https://ourworldindata.org/coronavirus-source-data>. Access is covered under the Creative Commons BY license. You have the permission to use, distribute, and reproduce in any medium, provided the source and authors are credited.

- selected countries (you will need to do some quick research on the countries you choose to include)
- iv. Populate these extra columns with the statistics you research for each country
 - v. Save this new dataset into a new .csv file which you can import into Python
- Create a Python scrip file with the name “*data_merger.py*”
 - This Python script **will use the Python petl framework** to:
 - i. Import your new CSV and “Country_location.xml” files into Python.
HINT: hard-to-find code to import this XML file using petl -
etl.fromxml('Country_location.xml', './tr', ('th', 'td'))
 - ii. Merge the country geolocation data from the XML file with the COVID-19 data, and
 - iii. Export the merged data as a .csv file named “*covid_countries.csv*”
 - **RESTful Web service server demo.** In this demo, you are required to build a RESTful Web service
 - This Web Service **requires the Python bottle and petl frameworks**
 - This Web Service server accepts a ‘getcountry’ query from the client browser similar to “/getcountry?country=xxxx”,
 - The server Web Service will return a JSON object to the client with all the attributes required by the client browser.
HINT: the data will be returned as a Json / Dictionary object -
etl.cut(datatable, "data", "data", "etc", "etc", "etc").dicts()[0]
HINT: in order for your code to return the JSON object to an AJAX request insert the following code before returning the JSON object –
response.headers['Access-Control-Allow-Origin'] = '*'
response.headers['Content-type'] = 'application/json'
 - The data for this JSON object will be retrieved from the “*covid_countries.csv*” file created in component 1.
 - A Python scrip file with the name “*country_locator.py*” will contain this RESTful Web Service server demo.
 - **Mashup demo.** In this demo, you are required to build a mashup application
 - Your HTML page accepts a country name as the input. If the country can be found in your dataset, then its location is displayed in Google Map. (See sample Mashup figure 1),
 - A HTML file “*country_map.html*” should be implemented.
 - In this file, a text field is provided at the page top to accept user’s input of country name.
 - In addition, a Submit button will trigger an AJAX “getcountry” Web Service GET query to retrieve the country’s corresponding latitude, longitude and other country specific data, and highlight and centre the location in Google map.

- In addition to highlighting the country's position on the map, a user is able to click on the marker bubble and see an information window containing country specific COVID-19 information and 2 to 3 other country statistics you have included in your dataset.

Important Note: For ease of demonstration to OSL executives, your code must be self-contained. In addition to using Standards based HTML and Python, petl, and bottle are the only frameworks you are will need. No other Python frameworks will be accepted.

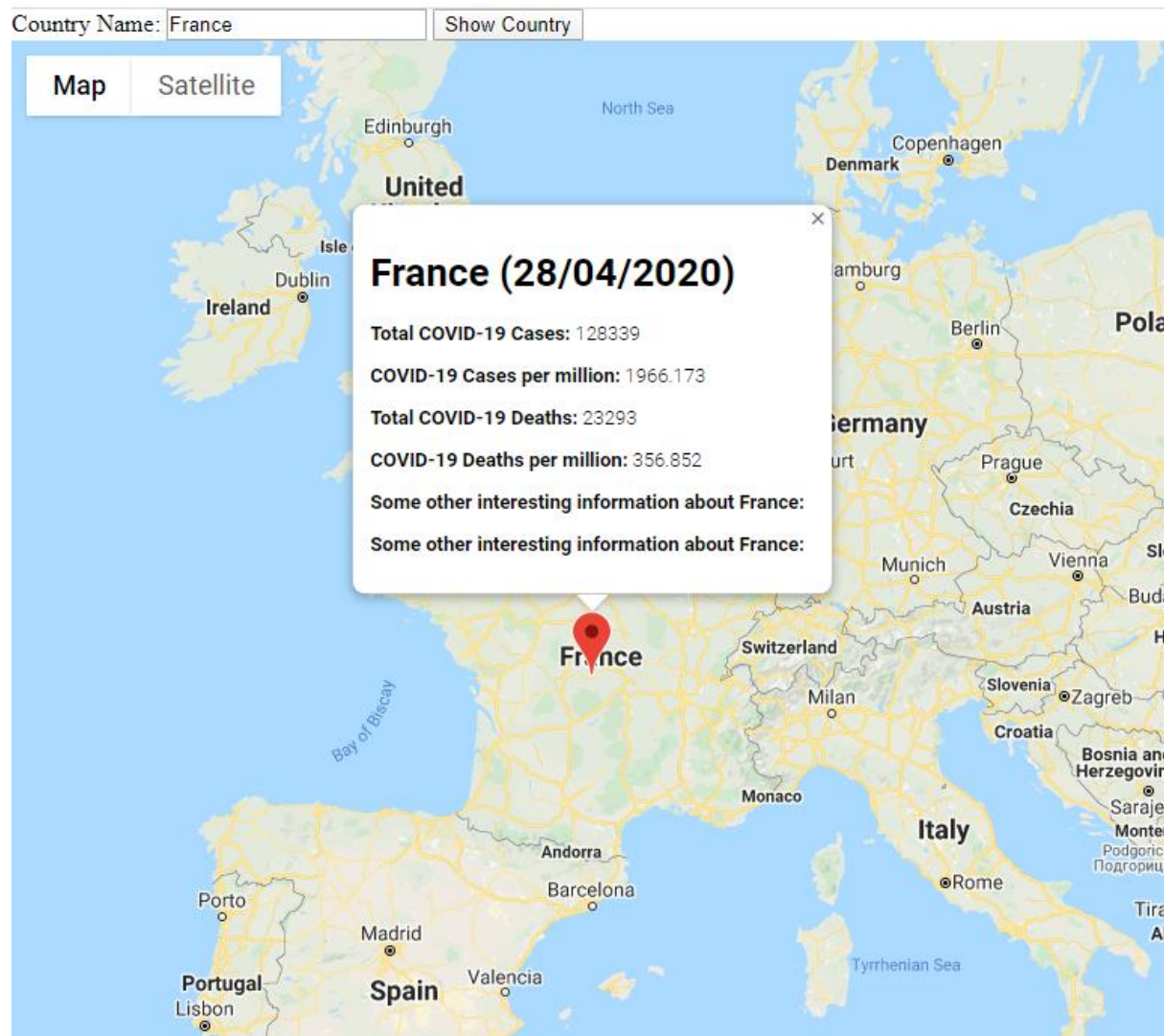


Figure 1 - Sample Mashup

Deliverable 2 – Integration Report (50%)

In addition to the above demo code you are required to include a Report explaining the key concepts around the demo design and implementation.

Your report should follow the following template:

Executive Summary

Table of Contents

Table of Figures (if needed)

Table of Tables (if needed)

1.0 Introduction

2.0 Service Oriented Architecture (SOA)

3.0 Key System Concepts

 3.1 Data Cleaning & Merging

 3.2 RESTful Web Services

 3.3 Mashup Design

4.0 Demo Running Instructions

5.0 Conclusion & Recommendations

Reference List

Appendices (if needed)

Please address the following in your report:

- Outline the theory and technologies under each of the headings.
- Explain how these technologies can be applied by OSL. Outline the difficulties, advantages and disadvantages to OSL. Give examples in addition to the COVID-19 application,
- Explain how to run your code and the principles behind each step – you may use screenshots of Mashups, etc. if you wish,
- Reflect on your journey as you developed the demonstration – the problems you faced, how you researched and found answers.

Report Format

Your report should be no less than 1500 words and it would be best to be no longer than 2000 words. Note that the code will not count towards the word count.

The report **MUST** be formatted using the following guidelines:

- Paragraph text – 12 point Calibri single line spacing
- Headings – Arial in an appropriate type size
- Margins – 2.5cm on all margins
- Header – Report title
- Footer – page number (including the word “Page”)
- Page numbering – roman numerals (i, ii, iii, iv) up to and including the Table of Contents, restart numbering using conventional numerals (1, 2, 3, 4) from the first page after the Table of Contents.
- Title Page – Must not contain headers or footers. Include your name as the report’s author.
- The report is to be created as a single Microsoft Word document (version 2007, 2010, 2013 or Office 365). **No other format is acceptable and doing so will result in the deduction of marks.**

Please follow the conventions detailed in:

Summers, J. & Smith, B., 2014, *Communication Skills Handbook*, 4th Ed, Wiley, Australia.

Submission

Submit your work as a zip file to Blackboard Task 2 by the due date of **11:59pm Friday, Week 12**. You are required to submit 1) Word file of report should be included and 2) a zip package of your code. No code submitted will result in a 0 for Task 2.

The assignment will be assessed according to the Rubric online. Late submission will be penalised according to the policy in the course outline. Please note Saturday and Sunday are included in the count of days late.

Assignment Return and Release of Grades

Assignment grades will be available on the course website in two weeks after the submission. An electronic assignment marking sheet will be available at this time.

Where an assignment is undergoing investigation for alleged plagiarism or collusion the grade for the assignment and the assignment will be withheld until the investigation has concluded.

Assignment Guidelines

This assignment will take a number of weeks to complete and will require a good understanding of application and information integration technologies for successful completion. It is imperative that students take heed of the following points in relation to doing this assignment:

1. Ensure that you clearly understand the requirements for the assignment – what has to be done and what are the deliverables.
2. If you do not understand any of the assignment requirements – Please **ASK** the lecturer or your tutor.
3. Each time you work on any aspect of the assignment reread the assignment requirements to ensure that what is required is clearly understood.