Enterprise Programming : Assignment Term 1 2016-17
Due: Friday 13 January 2017, 11pm

Developing, deploying, publishing and finding your own web service

In this assignment, you will create a simple service that is accessible in a variety of ways using different technologies, and critically assess these approaches.

You will develop a simple Courses info Web service from scratch, deploy it and develop a range of clients to access the data.

Interface

Your courses info API service will have the following (as a minimum) interface:

```
public interface CourseInfo {
  public void addCourse (CourseInfo cinfo);
  public Collection listCourse ();
  public Collection searchCourse(String searchStr);
}
```

The `Course` class can have attributes such as: courseID, courseName, courseCredits, courseDuration, courseTutor and others. You may amend the details the attributes as needed.

Storage

In order to implement the `addCourse`, `listCourse` and `searchCourse` operations, you should use a cloud database to store and retrieve staff records, though you may decide to use a local storage option (such as mysql) to test your web services initially.

You are to create a simple web based app to manage the above interface, and provide a variety of methods to access the data from a client app, as detailed below

Technologies to use

You should demonstrate proficiency in using the following technologies as part of your web service:

- Access to the data using simple http web service calls (10 marks)
- Options to return the data in text, json (the default) or xml (10 marks)
- Google App Engine or Microsoft Azure to implement the application on a remote cloud based server. (20 marks)
- A WSDL description of the interface to the web service (5 marks)
- Access to the data using REST type interaction (10 marks)
- An Ajax based web front end to retrieve the data and display in a suitable format using library based routines for an enhanced user interface (15 marks)

Evaluation

You are to give a critical analysis of your work and the techniques you have used. Here, you should evaluate the pros and cons of each approach, and design/implementation decisions taken. This section forms a major part of the assessment. You need to demonstrate that your code is professional, and follows sound Software Enineering techniques. You need to show where they are used in your code and why, giving an evaluation and the benefits of using these techniques. This will show your true understanding of both your code and the techniques used. (30 marks).

Deliverables

You should submit a zip (not rar or other formats) file containing the following documents:

- A PDF (not odt, doc or other formts) file showing your code working, using screenshots, and suitable descriptions. You should clearly indicate which sections you have completed.
- A README file describing the steps to invoke your programs for deploying, publishing and finding your Web service
- Commented code listings in the Eclipse project code (no need to show separately)
- An Eclipse project(s) with the code ready to compile. You may have more than one project for client and server if you wish. This is a good idea as you don't need one all-encompassing project that does everything – it's not realistic.

The single file should have a filename of your name and student number (surname first), e.g. WhittakerNick02386456.zip

Please note that the submission inbox on Moodle will not accept submissions larger than 100MB. Your zip file is unlikely to be this large unless you have used very large image files. Please check in good time that your work will fit within the size limit. If you need to go over 100Mb, provide a link in your documentation to the external site where you have stored the data.

CAUTION:
There are a number of online implementations of this and similar concepts. While it is OK to review these for ideas, it is not OK to copy whole or parts of these code examples and pass them off as your own. We will use an AUTOMATED PLAGIARISM CHECKER to compare your submissions against other students' work and online examples.

NOTES AND HELP
Assignment Discussion – Detailed description of what is expected

The learning outcomes of the assignment are that you should demonstrate that you know how to deploy a web service in several ways, that you can return data from a service in a variety of formats, and make use of cloud services to deploy apps.

You are demonstrating that you can create software that delivers these objectives. You will be developing a series of classes, effectively your own API, to offer a web service. Most of the assignment isn't concerned with the client front end, you are not writing software for the general public. You are writing software that other programmer can use, and for machine-to-machine communication. The front end will be some simple user programs that demonstrate the api is working, rather than something for a normal user.

In order to show that you can write a web service, you are going to offer access to a simple plain old java object (pojo). In this case, it is a Course object. You will offer simple access to store course details, retrieve courses and list courses. The objective here is to show that you can offer web services in a variety of formats, and return data in a variety of formats. We are deliberately keeping the actual object simple to reflect this.

The Course objects are to be based on a server, and there should be a collection of them. How the collection is stored will be up to you, but it should be cloud persistent in some way (ie not entirely memory based). Good solutions will implement a Data Accessor Object (DAO) to access the objects through a simple interface to the main server program. The DAO will hide the implementation of the persistence.

The final version should have cloud based server code, using Google AppEngine or Azure, depending on the implementation language of the server code. There are several methods of persistence, and you should choose an appropriate one. Remember that part of the marks is for your coding style and good use of software engineering techniques. You will have practice in using Google AppEngine and Datastores in lab sessions, and have reference to other methods.

The server code should offer a variety of methods to allow access to the data. Remember that you are implementing code for other programmers, so you are making it easy for others to invoke your web services.

You should implement a simple http type method of accessing the data using, for example, a servlet with appropriate parameters passed.

eg **`http://serverlocation/CourseServer/searchCourse?name=Mobile`**

This would return a list of courses containing "Mobile" in the name. The list of course details should be in some easily processed format, xml, json or structured text fields (using suitable delimitor). It should be possible for the user to select the format, with a default if none supplied, eg

`http://serverlocation/CourseServer/searchCourse?name=Mobile&format=json`
or
`http://serverlocation/CourseServer/searchCourse?name=Mobile&format=xml`

The server code will extract the data from persistent storage, and return the data in the prescribed format. Remember this is a web service returning just the plain data in a structured format. It is not sending it back formatted for viewing immediately. That is up to the client. You will look at that later with an Ajax based client. Make use of suitable libraries here to convert to/from json/xml/pojo.

You need to demonstrate that you can upload the data in json/xml format to, via a POST call. Data from the client will be wrapped in json or xml and sent to the server (eg for a new course entry). In the case of xml, this would probably be a SOAP call.

You should demonstrate calls to your web service in this manner, sending/retrieving different formats.

You are also to implement a RESTful interface to your server data, and demonstrate this working in a similar manner.

All you need to show is the http calls to the web service and the data received for all combinations. You need to include a wsdl to describe the web services for potential users (remember you can generate one of these automatically once you have your web service)

The final bit of coding is to create a client side app using Ajax type calls, which will invoke the web service in the same way as above, but take the response data and display it on screen in a user-friendly way in the browser. Make use of suitable libraries to help you here (eg jQuery). Show both upload and retrieval of data from meaningful user screens in a browser, using suitable widgets from your javascript library.

It will be useful to you, and less time consuming, to use suitable libraries and demos from lab examples that you should have completed. Remember that many parts of this can be auto-generated (eg a SOAP call using XML, which has an appropriate testing interface ready generated - see the lab when we did this for the temperature conversion example).

The final section, the critical analysis, is where you should discuss the code quality and use of suitable structures, why you made the design decisions that you did. E.g. is it MVC based (you should show it if so), do you use DAOs, is your code modular and easily amended. What other design patterns or techniques have you used? Look at examples of good SE practice and demonstrate that you are using them. Commentary on which api (SOAP, REST, http) is useful to different users, and why, is needed. Have you used the correct API for the task? Is your code easily maintainable? (and why?).

There will be extra assistance with the assignment in the form of an online guide on Moodle. You should also (if needed) discuss your approach with your tutor in labs, or personally after arranging an appointment at the availability times shown on the Moodle module home page.