TMA 01

Matthew Mason: C6122243

BSC (HONOURS) COMPUTING AND IT: TM-352

Table of Contents

Part 1: An Introduction to Web Services

web services are remote forms of functionality intended for computer-to-computer interaction, requested via a client using protocols and open standards (XML, HTML, HTTP, TCP/IP, etc) to exchange data between applications and different systems. They can be categorised as Read-Only, Write-Only or Read/Write and be approached in two different ways:

SOAP is a W3C standard ((W3C), 2007) defining how to construct, format and encode XML messages ((W3C), 2008), utilising application layer protocols for transmission over distributed environments. The framework consists of:

- <envelope> The mandatory root element of the hierarchical structure.
- <header> An optional element for adding attributes: role, relay, mustUnderstand.
- <body> Payload container for the ultimate receiver.

REST (Fielding, 2000), an architectural style, allows clients to utilise the HTTP operations GET, POST, PUT, etc, and use JavaScript Object Notation (ECMA, 2017), to access resources via a uniquely identified URI.

The two main data exchange methods for web services:

XML, a W3C standard, is data enclosed in tags that delimit, identify it and establish semantic meanings. Publicly available, application specific schemas and WSDL documents can be included in messages to provide clients an understanding of expected XML document structures, request, response tags and namespaces to avoid naming conflicts.

JSON is a lightweight, compact text-format based on a subset of JavaScript whose syntax closely maps to programming data structures, eliminating the need for a parser. Data is represented by name/value pairs with values representing various data types. JSON can utilise AJAX to transmit data without reloading a browser.

Advantages	Disadvantages
Firewalls can monitor traffic from requests and responses due to the use of existing protocols. Analysis of unique URI and HTTP methods can indicate intent.	Is not a standard but rather an architectural philosophy which doesn't provide a guaranteed level of reliability and formality.
Multiple text-based data methods can be used for requests and responses.	Is less secure than SOAP which supports Web Services Security for enterprise-level protection.
Is Stateless and doesn't require a constant connection between client and server.	Can only use HTTP whereas SOAP can utilise other protocols.
Efficient and similar to other web technologies in design philosophy making it easier to learn.	

Table 1: Advantages and Disadvantages of MegaMax using REST over SOAP

Based on the following considerations, I would adopt the REST approach to building MegaMax's web services:

The cost of production and maintenance wouldn't be expensive as the web service is relatively simple requiring only a small number or URIs to fully implement, REST programming languages are easy to learn and implement and no additional WSDL or application specific Schemas are required.

Connectivity to servers won't always be available, potentially requiring development of a customer facing website allowing clients to place orders themselves at a later date or MegaMax need to ensure staff devices are secure enough by using passwords or biometrics, etc, to store client data and prevent exposure if the device is lost or stolen. Secure data transmission and web service access can be achieved via simple server configuration of staff and single site SSL certificates.

(Word Count: 500)

Part 2: Security

There are numerous measures that can be taken to ensure the confidentiality and integrity of data during transmission from a remote location to MegaMax's servers. Authentication and authorisation methods, encryption algorithms, digital certificates and asymmetric key methods can be combined with the SSL protocol to exchange sensitive information over a secure channel.

SSL works using standard HTTP operations and negotiates between communicating parties to establish the parameters of data exchange. A client request directed at the URI of the web service hosted on the company server, initiates the conversation where an agreed encryption algorithm is decided between the two. These are chosen from pre-installed crypto-graphic service provides and their provided services and cipher algorithms, the most common being block ciphers such as Advanced Encryption Standard (AES) or Triple Digital Encryption Standard (DES).

The server replies with its public key; a method of encrypting messages that can only be decrypted by those with its equivalent private key, held by the sending party. The key could be as simple as adding 1 to each letter of the alphabet but generally pairs of keys are generated using extremely large prime numbers so even if one key is known it's infeasible to predict the other. The server also sends an SSL certificate. This is a digital certificate paid for by MegaMax that requires validation of the organisations public key, fully qualified domain name, and company details by a third-party Certificate Authorities Organization. It authenticates the organisation and ensures communications are secured and encrypted.

Following this, the client can utilise MegaMax's public key to generate an encrypted symmetric session key which provides a more efficient method of communication in terms of processing power and sends this back to the server who then decrypts it with their private key.

Once a secure connection is established, verification of sales staff to utilise the web service would require authentication by the client, via encoded user-identifiers and passwords. User details and authorisation would need to be configured on the server and after initial credentials have been confirmed, subsequent requests could contain a cryptographically generated OAuth 2.0 bearer token to authenticate the user. A method of checking the 'freshness' of the data must also be included such as a timestamp or a cryptographic nonce value specific to each exchange. These transmissions could also include hash values to provide message integrity and provide assurance the data has not been altered in any way.

The statelessness of REST ensures that no data is saved on the server but once processed, additional steps will need to be taken to secure access to databases where the processed information resides. Similarly, security measures will be required to prevent unauthorised access to remote devices, this could be in the form or biometrics or passwords which are not allowed to be saved to the device.

No security measures are perfect; however, the methods above should be enough to provide reliable and secure transmission and help counteract threats like denial of service attacks, message modification, fabrication and interception.

(Word Count: 500)

Part 3: Demonstrating A Web Service in NetBeans

To begin building the MegaMax web service, select Java with Ant | Java Web from the categories list and choose Web Application to create a new project.

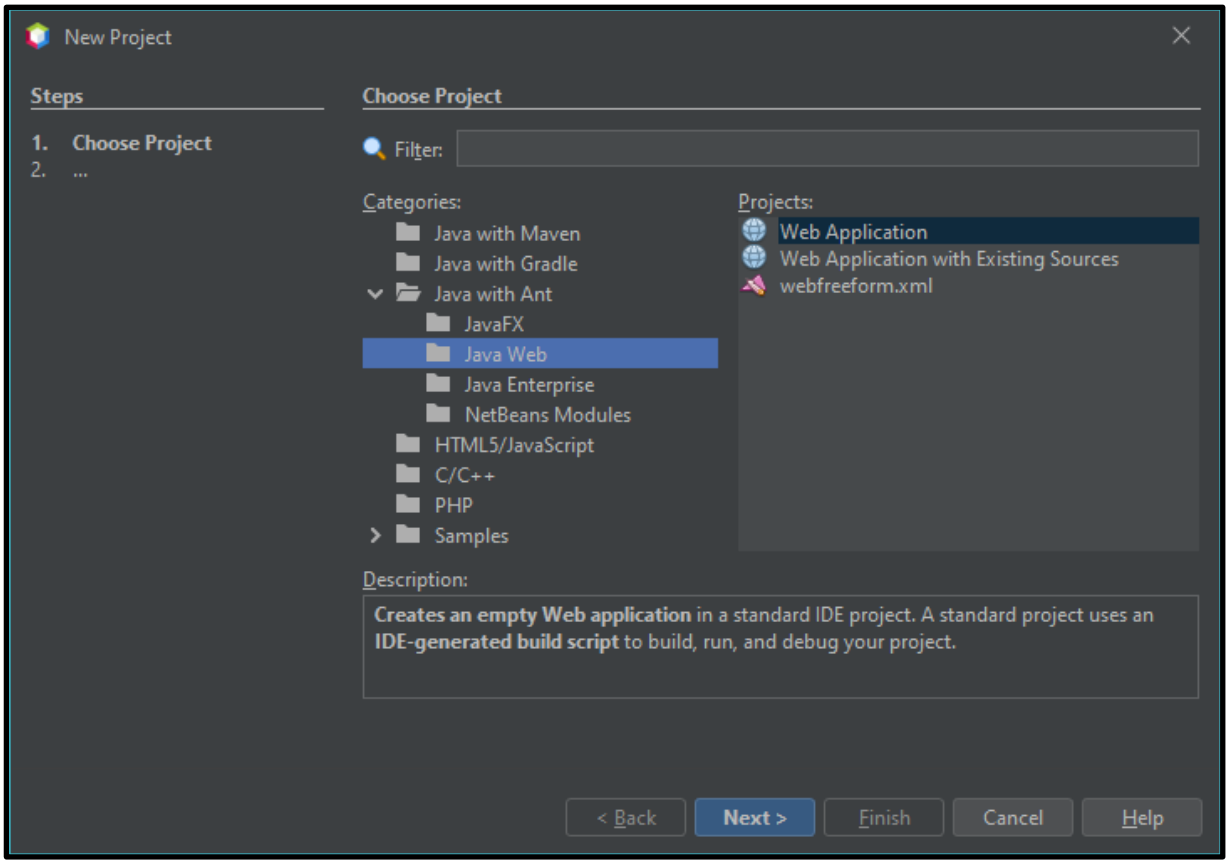

Figure 1: Creating a new Web Application

We establish the projects name and the location to which we will save our project.

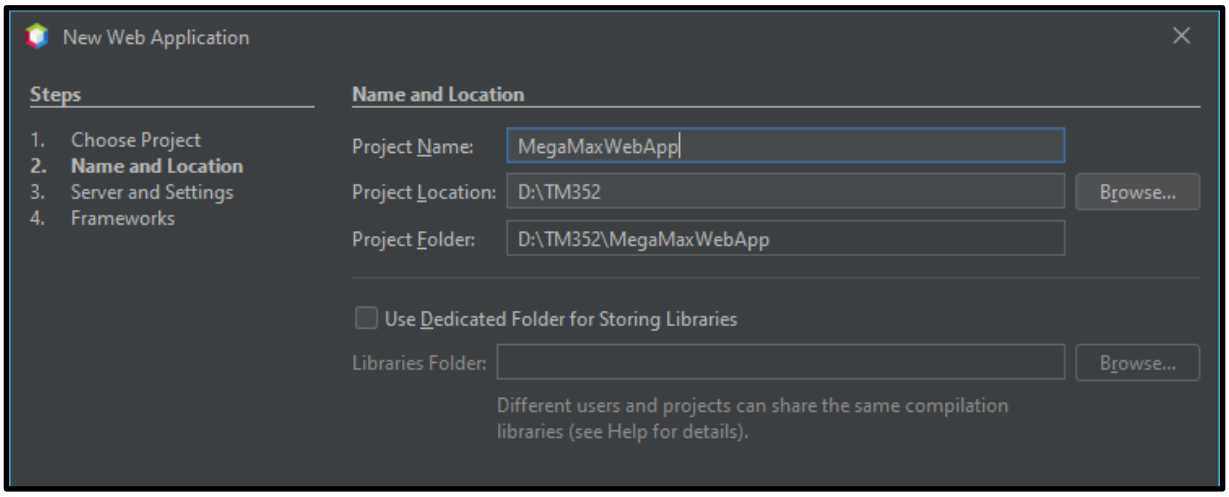

Figure 2: Name and saving the Project

Before we finalise creating the project, we have to confirm some simple server settings which should be filled in automatically. These include the type of server, in this case we will be using GlassFish, the Java EE version and the Context Path which is the root file of the URL and used to build a path to individual resources.

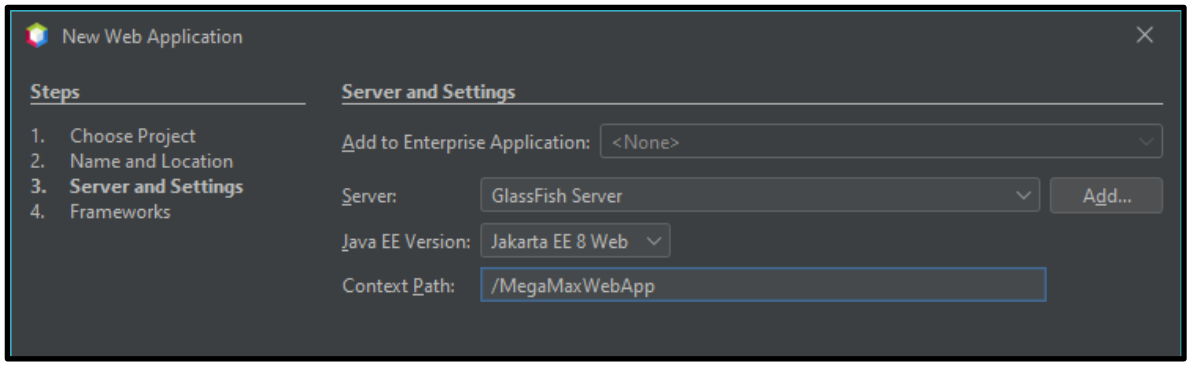

Figure 3: Confirming Server Settings

The Web service is a Java servlet, a server-side Java program that will process the data sent from the client. To add this, right click the project name in the projects tab, select new, then choose Servlet which provides us with the configuration options below. We fill out the name of the Servlet class, choose which project we will add it to and select a package for the class.

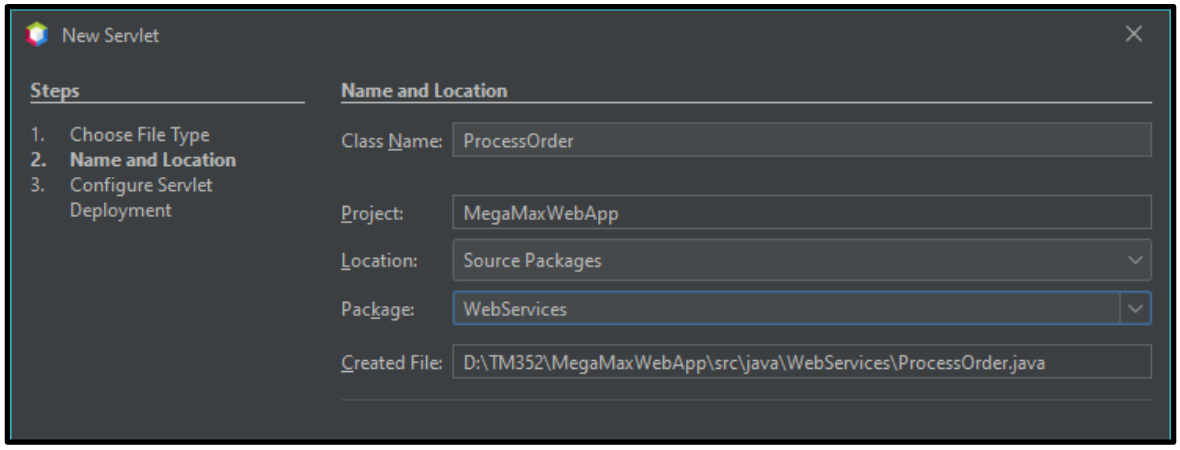

Figure 4: Adding the Servlet

The servlet provides us with some pre-written methods such as doGet, and doPost. These two methods simply respond to the different types of standard HTTP operations sent from the client and call the processRequest method which we will be editing to provide MegaMax with an example of how their own web service can operate.

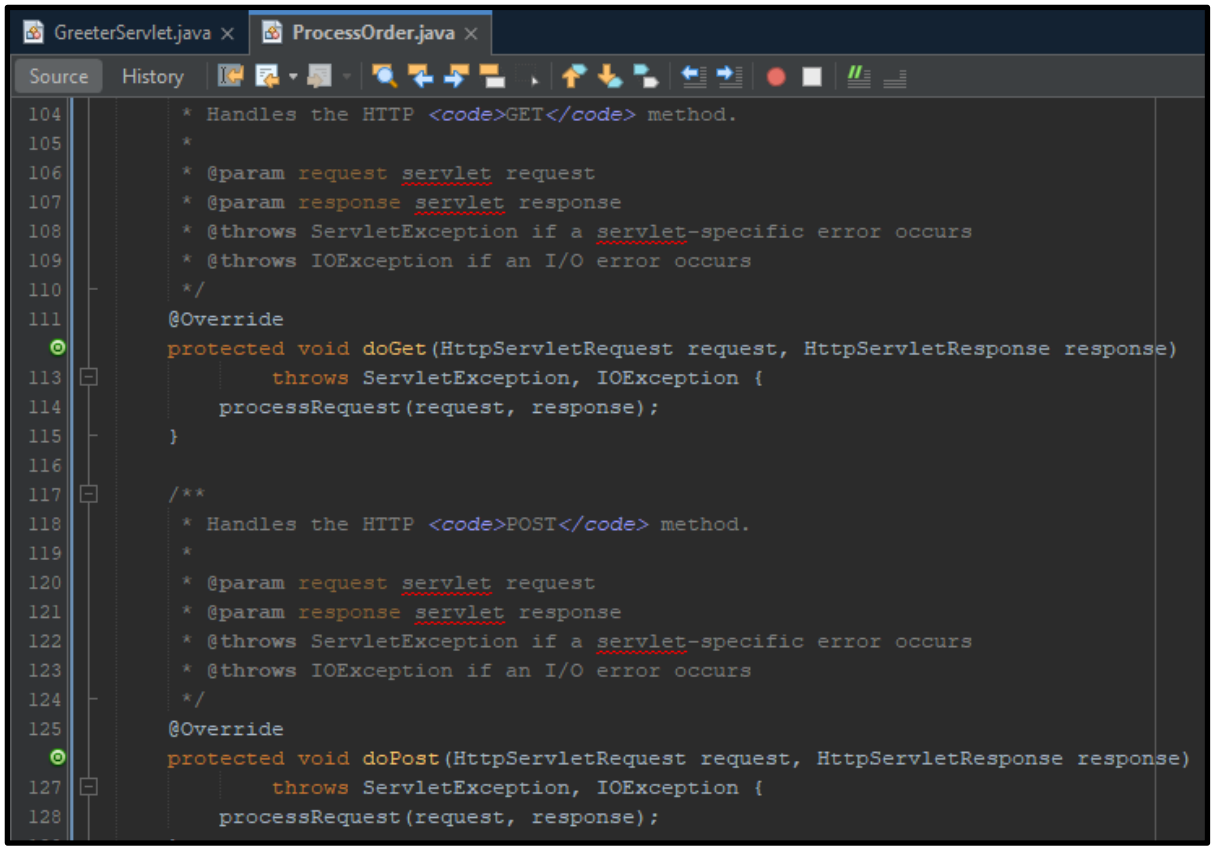

Figure 5: standard HTTP operation request handler methods

We must make sure to provide a relative URL for the project when it's first started to direct us to the correct page by building a URL from the root. In this case, we want to browser to direct to: http://localhost:8080/MegaMaxWebApp/ProcessOrder. This is set by opening the project properties, selecting Run from the category list and typing it in. This should match the URL pattern of the servlet class.

Figure 6: URL Pattern

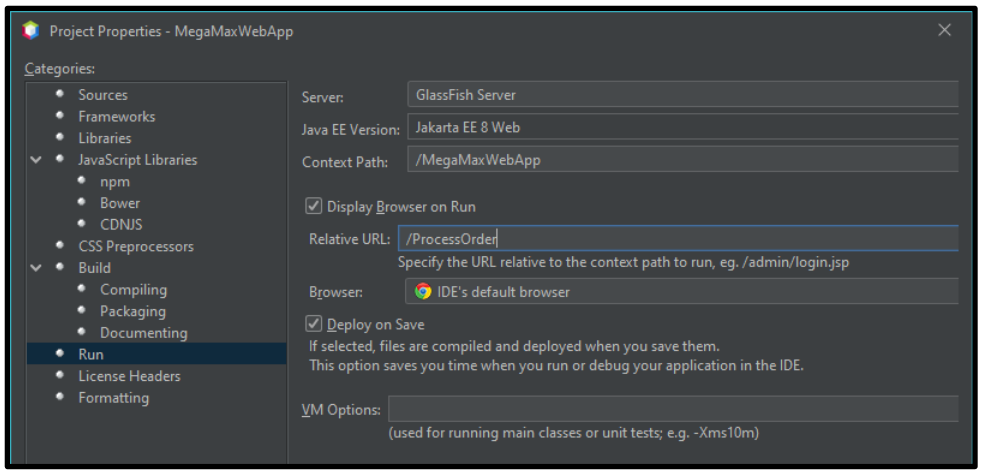

Figure 7: Relative URL

Utilising and adapting the processRequest method from Activity 1, Task 2, we can change the application to respond with a custom message. Copying the code prompts us to import a required package:

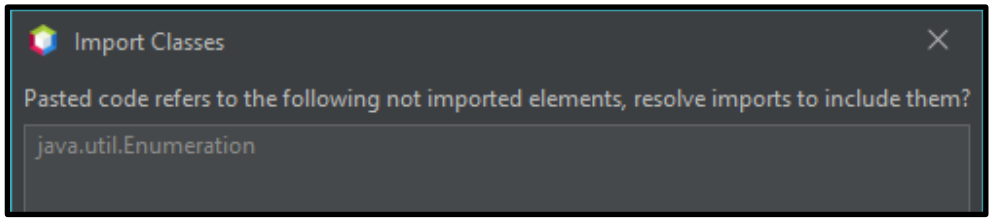

Figure 8: Import Java Class

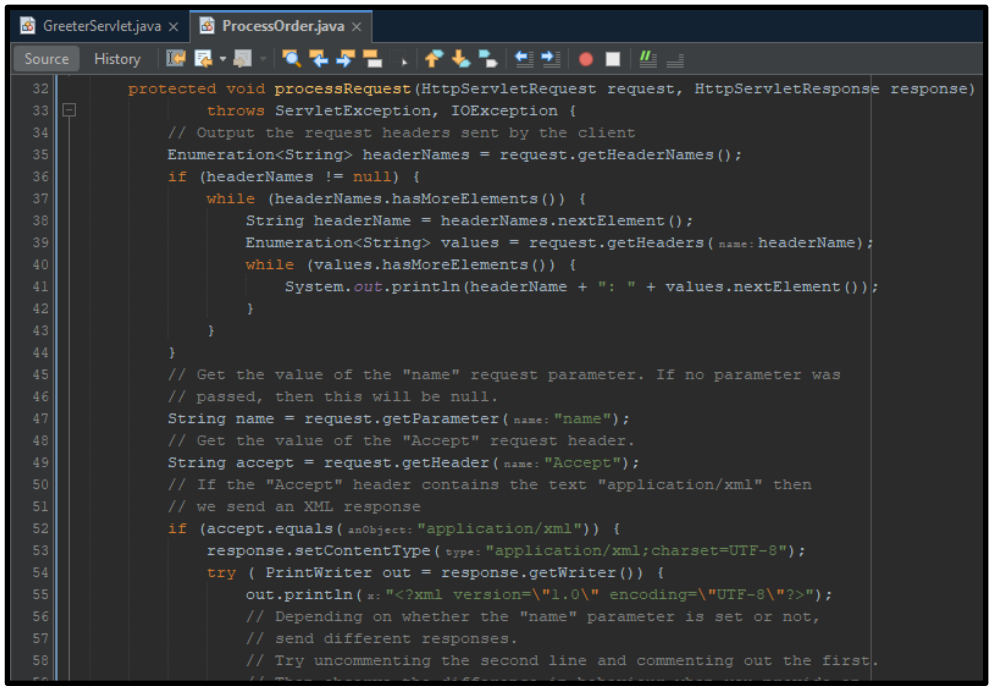

Figure 9: customised processRequest Method

When we run the project, the results should appear as follows:

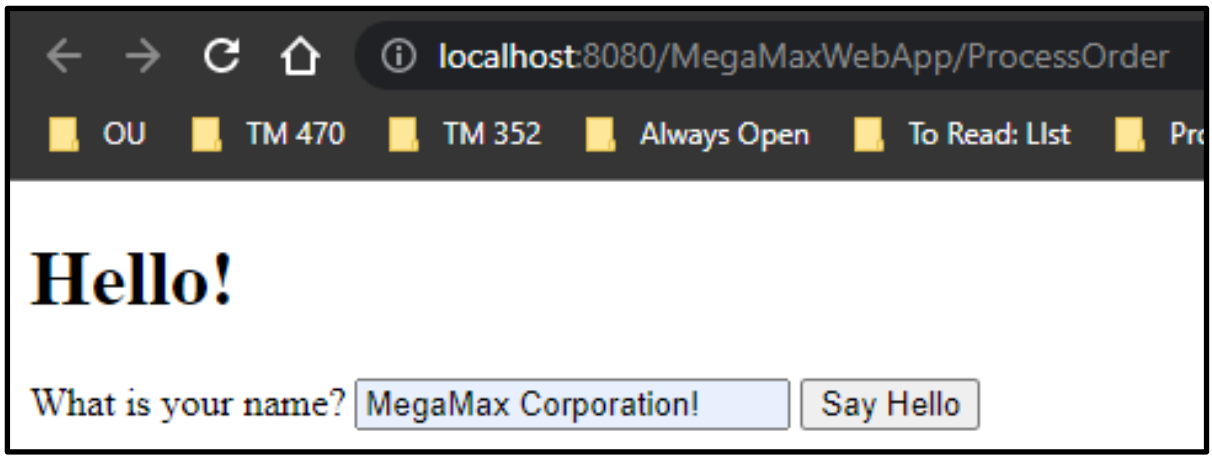

Figure 10: The Process Order Web Service

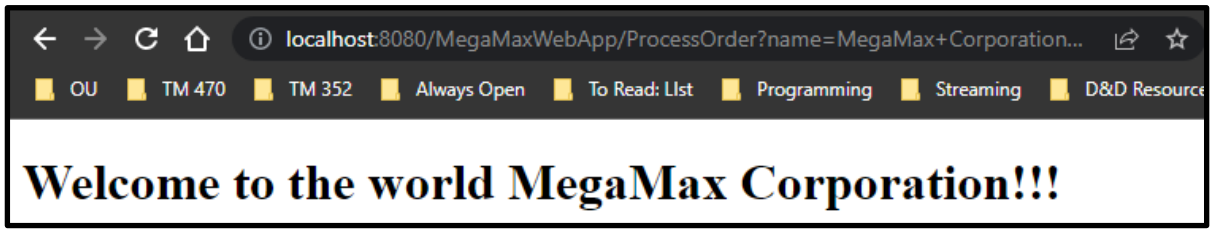

Figure 11: Response from the web service

(Word Count: 400)

References

(W3C), W. W. (2007, April 27th). *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*, 1.2. (M. H.-J. Martin Gudgin, Editor) Retrieved September 26th, 2022, from W3C: https://www.w3.org/TR/soap12-part1/

(W3C), W. W. (2008, November 26th). *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, 5. (J. P.-M. Tim Bray, Editor) Retrieved September 26th, 2022, from W3C: https://www.w3.org/TR/xml/

ECMA. (2017, December). *ECMA-404, 2nd edition, December 2017.* Retrieved September 26th, 2022, from ECMA International: https://www.ecma-international.org/wp-content/uploads/ECMA-404_2nd_edition_december_2017.pdf

Fielding, R. (2000). *Architectural Styles and the Design of Network-based Software Architectures.* Dissertation, University Of California Irvine, Computer Science. Retrieved September 26th, 2022, from https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

Team, C. (2022). *What is REST? | Codecademy*. (C. Team, Editor) Retrieved September Monday 26th, 2022, from Codecademy: https://www.codecademy.com/article/what-is-rest