Honours Degree in Computing

Web Services Assessment:
The changing landscape of web services

Submitted by: Derek McCarthy, B00007439

Submission date: 14/04/2019

Declaration

I/We declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, except where otherwise stated. I/We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged, and the source cited are identified in the assignment references.

I/We understand that plagiarism, collusion, and copying are grave and serious offences and accept the penalties that would be imposed should I/we engage in plagiarism, collusion or copying. I acknowledge that copying someone else's assignment, or part of it, is wrong, and that submitting identical work to others constitutes a form of plagiarism. I/We have read and understood the colleges plagiarism policy 3AS08 (available here).

This material, or any part of it, has not been previously submitted for assessment for an academic purpose at this or any other academic institution.

I have not allowed anyone to copy my work with the intention of passing it off as their own work.

Name: Derek McCarthy Dated: 14/04/2019

Contents

Introduction

The term web service can be interpreted as a collection of protocols used for passing and receiving data between two or more devices/pieces of software. It allows application using different programming languages to pass and receive data. This is achieved through the use of open standards. For example, a program application written in Java can interoperate with an application written in Python. The same principle applies for an operating system where a Windows machine can interoperate with a Linux machine and vice versa.

Protocols

A web service protocols main purpose is to provide a way of describing data and behaviour in a way computer/device can understand. In this section we will discuss the different protocols that are available today and disregard older protocols such as SOAP (Simple Object Access Protocol). The newer and more popular method of accessing web services is by using the RESTful/REST (Representational State Transfer) architecture. This method offers a lightweight loosely coupled services that can be accessed across the internet. It uses Uniform Resource Identifiers (URI) to send and receive requests from the client to the server and vice versa [7]. These requests are sent using the standard HTTP protocols. Security of a RESTful service can be handled by using Secure Socket Layer (SSL) or Transport Layer Security (TLS). It allows applications to be written in any programming language that can make web requests. There are several different frameworks that can be used to implement a RESTful web service. For the server-side frameworks RESTlet, Apache CXF or Spring Data etc can be used, for the client-side frameworks such as JQuery, Node.js and Angular etc can be used. RESTful web service can return data in either JSON (JavaScript Object Notation) or XML (Extensible Markup Language) format.

Other web service protocols that exist are Hessian, Universal Mobile Push Daemon amongst many more.

Hessian is a binary protocol that doesn’t require a large framework to be implemented. It’s a binary protocol that is designed to be interpreted by a computer machine rather than a human which would use a plain text protocol such as HTTP etc. It allows for the sending and receiving of binary data. It can be implemented in all the popular programming languages such as Java, C++, PHP, Python and Ruby etc [10].

Universal Mobile Push Daemon (Pushd) is pluggable web service that can be used for sending server-side push notifications to mobile devices and web applications. It use’s a single-entry point to send these notifications to mobile devices and web applications. The user does not need to worry about which devices are subscribed to what events on the Pushd server as Pushd takes care of this and supports and unlimited number of subscribers for any service event [11]. It has several features such as Broadcast receiving, Direct Messages and Server-side message translation etc. Pushd also supports all the popular mobile operating systems and their protocols for sending and receiving notifications such as iOS, Android and Windows phone etc.

Security

To ensure security of a RESTful web service communication end points must use HTTPS. HTTPS establishes a secure link between two applications/devices by using TLS or SSL to guarantee that data stays protected during transfer. By using this security mechanism this protects data in transfer such as “passwords, API keys and JSON Web Tokens” [1].

Private REST services are required to perform access control at either end of the web services API communication end point. In single-tiered applications this can be achieved by the authentication of users and session management. To allow for a more decoupled system and minimise network latency, access control decisions should be made locally at either end of the REST web service endpoint. To authenticate users an Identity Provider (IdP) should be used. An IdP is a service that verifies users by issuing access/security tokens which are validated and verified by the IdP provider mostly through the use of OAuth [2].

Recently the use JSON Web Tokens (JWT) has seeing increase in usage as the preferred method for security tokens. JWT is a self-contained technique used to securely transmit data between two applications/devices using a JSON object. The data is validated and verified by its digital signature or message authentication code (MAC), which can be secured by using HMAC algorithm, RSA or ECDSA [3].

Private applications that use REST services without some kind of access restrictions are in jeopardy of running up huge bills for excessive bandwidth usage. By implementing API keys in the web service, you alleviate the risk of such cases. An API key is an application programming interface (API) that is used to identify the program using the web service, the developer of the program using the service or the user of the service. Using API keys also protects against the possible threat of denial of service (DNS) attacks.

Other ways of securing a web service is to restrict HTTP methods. This involves creating a list of allowed HTTP methods such as POST, GET and PUT etc and only allowing requests that have the same HTTP method that is stored in the list. The validation of input forms should also be carried out on all input fields to minimize the risk of security attacks or breaches.

Implementations

As there are many different implementations of web services available today. In this section we are going to focus on and discuss how to implement the RESTful Swagger API. Swagger is the most popular tool for developing an API for the OpenAPI Specification (OAS). It has a vast number of tools which can be used for both open source or professional projects [4].

There are two ways in which swagger can be implemented, from scratch or from an existent API. If you are starting from scratch, you first create the required OAS definition using the Swagger editor. The swagger editor allows you to specify the

required components you will need to build your API. You can choose between two types of schemas HTTP or HTTPS. It allows you to use either the lightweight JSON or the more cumbersome XML to transport data. Once you have specified and built the API you can then generate the server or client code depending on what you require. This code can be generated into a variety of different programming languages such as Java, Kotlin, R, PHP and python etc. Once you have generated the swagger definition code you then download the Swagger Codegen. Which handles the server-side implementation. Swagger also provide a user interface which allows you to interact visually with the API without the logic code. Creating API documentation is made easier and more intuitively with Swagger Hub. It allows you to collaborate with other developers and store templates of styles which can be reused at later dates.

Unlike other web service APIs where existing APIs need to be reversed engineered to implement into you project. Swagger Core makes this task easier by allowing you to create an OAS from an existent API. This allows you generate the OSA from your existent Python/Java API etc. It also supports several different frameworks such as Node.js or JAX-RS. If your API already has an OpenAPI definition, then you can use the Swagger Inspector to inspect the API. The Swagger Inspector is an online tool which allows you to analyse an existent API. Once your happy with the API you can use the Swagger Codegen to generate the required client code such as Java, PHP and Python etc.

Performance

In this section we will discuss the potential causes of poor performance of a web service. Poor performance of a web service can be caused by slow response times and slow servers hosting the API [5]. To avoid this network bottlenecking and potentially save money in your service a performance testing framework should be implemented. This always allows your web service to run at its optimal performance.

There are several different types of performance testing that can be used depending on your web service. The types of these tests are Load testing, Stress testing, Endurance testing and Custom test cases. As not all web services are the same choosing the correct type of test is paramount to ensure your test provide the most accurate data.

Load testing is useful for determining how your web service performs when a number of users access the web service. Typically, when using a load test the number of requests for the duration of the test increases, but this allows you to gather performance data from the test however big or small it was. Stress testing is almost identical to load testing except stress testing is designed to test the web service at its maximum capacity. This allows you discover at what capacity your web service is likely to crash and how the service performs at its maximum capacity. Endurance testing is used to test a web service over a prolonged time frame, which regulates the services endurance, and how the service behaves. This endurance testing is carried out on the web service during typical use. Custom test case allows you to create test cases based on user interactions with the web service.

Cloud based solutions

Cloud based web services are services that are made to users over the internet rather than the web service being stored locally. One of the main advantages of using cloud-based solutions is that they are scalable. This means you can increase/decrease your service requirements at any time rather than have a fixed requirement at all times. Other advantages of such services are they allow easy access to web service resources and services and can be managed from any location over the internet. There are several different companies offering cloud-based web services such as IBM Cloud, Amazon Web Services, Microsoft Azure and Google Cloud [6].

IBM Cloud offers a variety of different services such as virtual-based or hardware-based. Which can be used as public, private or management networks. Both the virtual-based and hardware-based servers are bundled into one cloud-based platform. This provides the users with full infrastructure control of the servers. Also, unlike other cloud-based service providers IBM's hardware-based servers users have sole control of the server which improves the performance of the services as the server is not being shared between users. Also, IBM offer full server customisation which allows the users to choose the features they require and not ones they do not need which cuts costs as your not paying for features you do not need.

Amazon Web Services offers a variety of different services either on the Infrastructure as a service (IaaS) or Platform as a service (PaaS). These services include the "Elastic Cloud Compute (EC2), Elastic Beanstalk, Simple Storage Service (S3) and Relational Database Service (RDS)" [6].

With Microsoft Azure you can run any service on the cloud or you can combine it with your current project service. They offer over 100 different services that can be used in your application such as Business SaaS (software as a service) apps, big data analytics, digital marketing and E-commerce etc. Azure is located in 54 regions worldwide more than any other cloud-based service provider and is available in 140 different countries.

Google Cloud similarly to Amazon Web Services offers vast amount of different modular web services which include the IaaS and PaaS web services. Applications developed on Google cloud are secured using a multi layered secure infrastructure. Which gives its users the confidence that the applications created/stored on its servers are protected at all times. Some of the web services offered by Google include App Engine, Compute Engine, Cloud Identity, Cloud Search, Business Intelligence and many more.

References

[1] https://github.com/OWASP/CheatSheetSeries/blob/master/cheatsheets/REST_Security_Cheat_Sheet.md

[2] https://www2.empowerid.com/learningcenter/technologies/service-identity-providers

[3] https://jwt.io/introduction/

[4] https://swagger.io/tools/open-source/getting-started/

[5] https://assertible.com/blog/web-service-performance-testing-tips-and-tools-for-getting-started

[6] https://www.techradar.com/news/best-cloud-computing-service

[7] https://javaee.github.io/tutorial/jaxrs001.html

[8] https://searchmicroservices.techtarget.com/definition/REST-representational-state-transfer

[9] https://github.com/node-modules/hessian.js

[10] https://github.com/node-modules/hessian.js

[11] https://github.com/rs/pushd