

IT DIPLOMA 3.1 – Create and Develop REST APIs



Diploma of Information Technology

SUBJECT 3: Back End Web Development

Module 1: Create and Develop REST APIs

Copyright

All material in this document or within this section of published material on the Equinim Hub is copyright © Equinim College 2024 except as outlined below, and may not, in whole or in part, and in any form or by any means (electronic, mechanical, micro copying, photocopying, recording or otherwise) be reproduced, stored in a retrieval system, published, transmitted or altered in any manner without prior written permission of the copyright owner.

Version #01-20240111

LESSON 1: THE WORLD WIDE WEB



Introduction




How the Internet Works

LESSON 2: FEATURES AND DIFFERENT APPLICATIONS THAT APPLY TO THE HTTP NETWORK PROTOCOL




What is the HTTP network protocol?

 Features and different applications of the HTTP protocol

 What is HTTPS?

LESSON 3: FEATURES AND ANATOMY OF REST API HTTP

 Features of REST APIs

 Anatomy of REST APIs

LESSON 4: LANGUAGE USED IN PROGRAMMING


 Introduction


 JavaScript

 Java

 Python

 PHP

 C and C++

 Go (Golang)

LESSON 5: OPERATIONS METHODS AND FEATURES

 Operations

LESSON 6: CORS METHODOLOGY

 What is CORS Methodology?

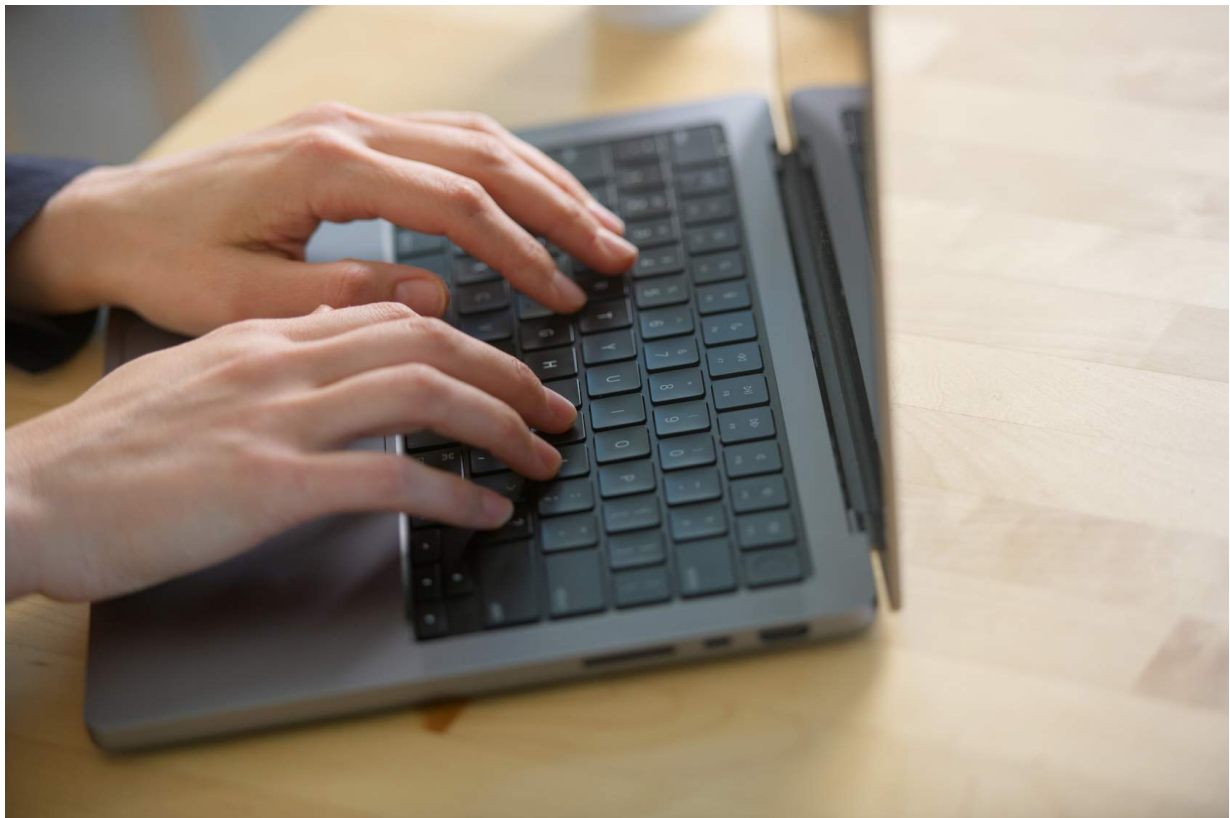
 **How does CORS work?**

ASSESSMENT

 **Assessment**

Introduction

If you're new to learning about information technology, then there are a few bits and pieces you need to understand before beginning this unit. This unit is all about creating and developing REST and APIs – however before these are explained, you need to understand how the World Wide Web works.



How the Internet Works

The internet is a computer network that stretches worldwide, transmitting data and media across devices that are all interconnected.

The internet uses something called a packet routing network, that is following an Internet Protocol (IP) and Transport Control Protocol (TCP). You may have heard of an IP address before, and that often this is how unwanted cyber behaviour is traced. TCP and IP work side by side to make sure that data is transmitted reliably. IP is essentially a bunch of rules that govern how information gets from one place to another over the internet. When data travels across the web, it is delivered in what is referred to as messages and packets – The message is the data, and the packets are the sub-parts of this message, broken up and made smaller.

How does the internet work?

Let's follow a real-life example of this at play. Imagine you would like to visit a website and follow through the following steps.

Step 1: Request

You connect to the internet, and type in the web address, known as a URL (Uniform Resource Locator), this signals to your ISP (internet service provider) where you would like to go. Chances are that the website you typed in uses HTTP at the beginning. HTTP is a protocol that uses a client-server request (more in Lesson 2).

Step 2: Connection

Your query goes to your ISP, and you are connected to lots of different servers which store and send data. Your browser then looks up the IP address and domain name that you typed in which is then translated into a number-based IP address.

Step 3: Request is sent

The browser then sends a HTTP request to the correct server to send a copy of the website using the TCP/IP that we discussed earlier.

Step 4: Request Approved!

The server then approves the request and sends a confirmation message to your computer, which will be in the form of a “200 OK” message. The server will then send files to your browser as data packets.

Step 5: Loading...

Finally, your browser will reassemble the data packets as the website loads.

Summary

So as you can see, there is a lot going on for a simple request and response that we use every day on our computers!

What is the HTTP network protocol?



As was just mentioned before, HTTP is a protocol. A protocol is a set of rules that apply to formatting and processing data. The HTTP protocol is a client-server protocol, which means that a client makes a request, which is initiated by the recipient, which in most cases is the web browser.

When you write a website into a web browser bar, you are initiating the request, a document is then reconstructed from sub documents to give you the web page you wish to view – which is the response from the server.

Features and different applications of the HTTP protocol

Features of the HTTP network protocol include:

- It is a client-server protocol
- It allows web servers and browsers to exchange information and data over the web
- It uses TCP connections on TCP port 80. If you recall, TCP stands for Transmission Control Protocol. Port 80 is the port number assigned to HTTP – it is the default network port used to send and receive web pages that are unencrypted
- It is stateless, which means that the client and server are aware of each other during a request only, but this is forgotten once the request is complete
- It is media independent, meaning it specifies that media content can be sent by it, so long as the data content can be handled by both the server and the client
- It is connectionless, meaning the client (the browser) makes the request and after the request has been completed the client disconnects from the server and awaits the response.

- It is simple and human readable, meaning easier testing for developers

Different applications of the HTTP protocol include:

- Initiating server requests and receiving responses.
- Controlling caches – A cache is software or hardware that is used to store data temporarily in a computing environment. When you visit a website, your browser will save certain resources from that website, such as data and/or images in a place called a cache. Cache control is a HTTP header that alters that browsing cache behaviour.
- Enabling authentication – HTTP can provide basic authentication for web pages.
- HTTP can enable sessions using HTTP cookies. Cookies are essentially collectors of information. They are text files containing pieces of data. Cookies allow websites to remember you, such as website logins, shopping carts and more.
- HTTP cookies specifically are there to identify you and therefore improve your internet browsing experience.

What is HTTPS?



Hypertext transfer protocol secure (HTTPS) is a secure version of HTTP, as is now the primary protocol that is used to transfer data between a web browser and a website.

The increased level of security in HTTPS, as opposed to HTTP is that it is encrypted, which is a huge advantage particularly when sending sensitive information, such as banking details, emails etc. Commonly used web browsers, such as Google Chrome, classify HTTP websites as non-secure and flag as such, HTTPS is therefore becoming the more commonly used protocol.

Features of REST APIs

API stands for application programming interface. This defines the rules that need to be followed in order to be able to communicate with other systems.

REST stands for representational state transfer. REST is a software architecture that demands how an API should work, i.e. how different applications communicate with one another. In REST APIs users are allowed to create, update, retrieve and delete data items.

The 4 basic methods, also referred to as CRUD operations include:

- 1 Create
- 2 Retrieve
- 3 Update
- 4 Delete

In a HTTP request, which we covered in Lesson 2, the above CRUD methods are used to use, access and sort data, in a way that is easy, simple and efficient. Therefore, HTTP is the application protocol, whereas REST is the set of rules that need to be followed e.g. all functionalities need to be described using certain functions, such as

CRUD, as touched on above. The most commonly used features will be covered in lesson 4.

REST API, also called RESTful API, therefore is an interface that two systems use to exchange information over the internet. Additional features of REST APIs include:

- Code on demand – In REST architectural style, servers can customise client functionality. This is done by transferring programming code to the client.
- Layered system – This refers to the ability to run several servers with multiple layers such as application and security working all together, however these layers are invisible to the client.
- Uniform interface – So that information and data will be transferred in a standard form.
- Data is cacheable and client-server interactions are streamlined.
- Client-server communication is stateless, meaning that client information is not stored or remembered between requests.

```

<li><a href="index.html">Home</a></li>
<li><a href="home-events.html">Home Events</a></li>
<li><a href="multi-col-menu.html">Multiple Column Menu on Larger Viewports</a></li>
<li><a href="#" class="current">Header Options</a></li>
<ul>
<li><a href="tall-button-header.html">Tall Button Header</a></li>
<li><a href="image-logo.html">Image Logo</a></li>
<li><a href="tall-logo.html" class="active">Tall Logo Image</a></li>
</ul>
<li><a href="#" class="has-children">Carousels</a>
<ul>
<li><a href="variable-width-slider.html">Variable Image Width Slider</a></li>
<li><a href="testimonial-slider.html">Testimonial Slider</a></li>
<li><a href="featured-work-slider.html">Featured Work Slider</a></li>
<li><a href="equal-column-slider.html">Equal Column Slider</a></li>
<li><a href="video-slider.html">Video Slider</a></li>
<li><a href="mini-bootstrap-carousel.html">Mini Slider</a></li>
</ul>
</li>

```

Anatomy of REST APIs

The anatomy of a REST request is as follows:

1

HTTP method – this describes what needs to be done with a certain resource – this is the CRUD operations that we touched on earlier.

2

Endpoint – This is the URL that you requested.

3

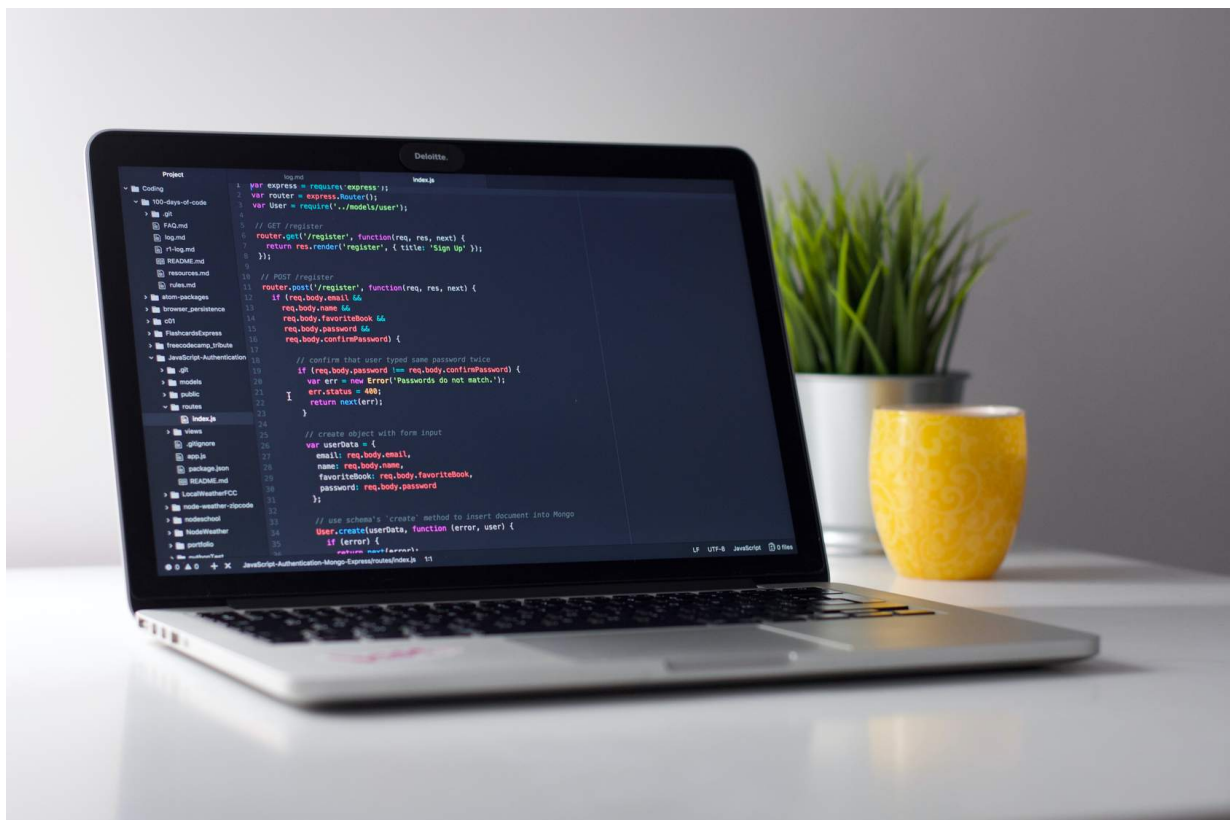
Headers – Headers store information which will be important to both the client and the server. Headers provide authentication data such as an IP address of the computer where the server exists, the name or something called an API key, as well as information regarding the response format.

4

Body – This is used to convey extra information to the server, such as a piece of data that you are wanting to replace, add or change.

Introduction

Programming really is like another language, with a vast array of words and symbols that you will be learning which may appear foreign at first but will eventually take on a simple meaning – much like reading English for the first time! Programming language is the way that a programmer communicates with a computer. Typically, this language will contain a set of rules that allows string values to be converted into graphical elements, or specific ways of generating machine code.



In the programming world however, there is more than one language depending on the type of program you are wanting to use or the one you see the most value in learning (in the case of wanting employment in a specific company that may favour one language). You may find value in learning more than one depending on your career and needs. A basic description of each will be covered in the following lesson.

JavaScript

JavaScript is one of the core technologies of the World Wide Web and one of the most popular programming languages currently in the world – currently being used by an estimate of around 97.8% of websites.

JavaScript is known as a scripting language, and is responsible for controlling multimedia, animating images, and basically anything else you see on websites that is moving, interactive or updating.

JavaScript can:

- Build websites containing interactivity
- Build web applications containing interactivity
- Be used to build games and gaming applications
- Animate images on websites
- Control multimedia
- Work with AI (artificial intelligence)

Java

Java is a general-purpose programming language with an object-oriented structure. Java can be used on all platforms because of its Write Once, Run Anywhere (WORA) capabilities. There are approximately 3 billion devices that employ the use of Java.

Java can:

- Be used in web and application development
- Be used in big data
- Be used in the backend of several highly popular websites including YouTube, Amazon, Twitter and Google

Python

Python is also an extremely popular programming language and is also praised as being relatively easy for beginners, as it uses a simple syntax that is similar to English.

Python can:

- Create web applications on a server
- Handle large amounts of data and perform mathematics
- Be used alongside software
- Connect database systems
- Read and modify files
- Be used for software development

PHP

PHP is an open source general-purpose scripting language, which can be embedded into HTML and is well suited for web development. One of the essential programming languages to learn, as it has been used to build more than 80% of websites on the internet. PHP is regarded as relatively easy to learn for beginners.

PHP can:

- Write server-side scripts
- Write command-line scripts
- Develop desktop applications
- Develop dynamic page content
- Collect form data
- Encrypt data

C and C++

C and C++ is known as a cross-platform language that can be utilised to create applications with high performance. It is one of the oldest and most popular programming languages around today. C is known as the root of other programming languages such as JavaScript, Java and C#. C++ is simply an updated and enhanced version of C.

C and C++ can:

- Make applications that are highly transportable due to the use of compilers for a variety of platforms
- Develop browsers
- Develop operating systems
- Develop applications
- Be used in in-game programming
- Be used in software engineering
- Be used in data structures

Go (Golang)

Go is an open-source, statically typed and compiled language designed by Google. It was developed originally for APIs and web applications and is currently one of the fastest growing languages due to its immense simplicity and abilities, including great speed. Go can typically handle massive codebases, as well as multicore and networked systems.

Go can:

- Carry out programming for large software systems
- Handle heavy requests on the server-side without using much memory
- Distribute network services
- Be used for utilities and stand-alone tools
- Be used for Cloud-Native development
- Build web applications

Google

Google Search

I'm Feeling Lucky

Operations

The following is a brief description of some of the more commonly used functions in REST APIs.

Operations Methods and Features

The following is a brief description of some of the more commonly used functions in REST APIs.

HTTP GET

Using a GET request will retrieve a resource representation/information only – unmodified. It is important to note that a GET request will not change the resource's state in any way.

POST

POST will create a new resource into an already existing collection of resources, at the specified URL.

PUT

PUT exists to update an existing resource.

OPTIONS

Represents a request for information about communication options available, which helps a client to determine the capabilities, or requirements of a server or resource. Seen in CORS (covered in Lesson 5).

What is CORS Methodology?

CORS stands for Cross-Origin Resource Sharing. CORS methodology is a HTTP based protocol that allows you to make requests from one website to another site in the browser, which is typically not allowed by another browser policy known as Same-Origin Policy (SOP).

To better understand CORS Methodology we first will cover Same-Origin Policy (SOP). SOP was invented to prevent the malicious activity that was occurring on the internet where certain websites were able to exploit cookies to make unauthorised requests to other domains – imagine someone being able to make website purchases or look up bank details all through your browser without you even knowing about it! SOP was invented to be able to prevent this happening by detecting when a request was being made from another website and making the response unreadable. SOP dictates that when a browser loads a script from a particular domain, that the script can only make requests to the domain that it originated from.

CORS Methodology therefore, allows servers to have certain trusted origins that they are happy to permit requests from.

How does CORS work?

CORS takes place when a script loaded from one origin makes a request to another origin. When this takes place, what occurs will depend on the type of requests, however it is important to note that there are specific HTTP headers that need to be included in the request and response of the other server, before a request can be allowed.

There are three main types of requests that CORS deals with:

1

Simple requests – such as GET or POST, which we touched on in Lesson 4. These simple requests will only send CORS safelisted Headers, which will trigger the interaction with the other server to respond with the requested data.

2

Non-simple requests – when a request is non-simple, the browser will send a preflight request before the actual request. A preflight request checks whether the request will be allowed by the other server. If you recall in Lesson 4 we briefly touched on the OPTIONS functions – A preflight request is an OPTIONS function as it uses three HTTP request headers. The preflight request lets the server know which HTTP method is used, and if any custom HTTP headers are present. The server can then decide whether it will accept a request from this origin or not. If it accepts it will send headers that will indicate this and then the actual request will take place.

Credential requests – This is a specific request regarding cookies, TLS client certificates or authorization headers. Typically, CORS does not allow credentials in cross-origin requests unless it is requested by the server and the other server responds correctly – this would be a setting that would need to be implemented into the server.

Assessment



SUBJECT 3

Back End Web Development

Module 1

Create and Develop REST APIs

Well Done!

You have now completed the **LEARNING** for this section.

What's Next?

You can now undertake the **ASSESSMENT** for this section.

Click the **RETURN** button at the top right of your screen to go back to your Module Home Page where you will find further details.