

PLAYING IN A SANDBOX WITH AUTHENTICATED SECURITY

(700100)

*700100: REPORT*

*TRUSTWORTHY COMPUTING*

*18 MAY 2020*

Farah Aly - 201505395

Contents

[Introduction 1](#_Toc40622692)

[Report 2](#_Toc40622693)

[API 2](#_Toc40622694)

[*Definition* 2](#_Toc40622695)

[*How an API works* 2](#_Toc40622696)

[*API response* 2](#_Toc40622697)

[Inside the PayPal Sandbox and REST API 3](#_Toc40622698)

[Weakness 4](#_Toc40622699)

[*Dummy Catering System* 4](#_Toc40622700)

[*PayPal Sandbox* 4](#_Toc40622701)

[Short Note 6](#_Toc40622702)

[Authenticated Encryption Modes 6](#_Toc40622703)

[Generic Composition Methods of Authenticated Encryption Modes 6](#_Toc40622704)

[Single-Pass Authenticated Encryption Modes 6](#_Toc40622705)

[Conclusion 7](#_Toc40622706)

[References 8](#_Toc40622707)

[Appendix 9](#_Toc40622708)

[Appendix A: Pseudo Code 9](#_Toc40622709)

[Appendix B: Tests 10](#_Toc40622710)

[Appendix C: Heuristic Evaluation Results 1](#_Toc40622711)

# Introduction

The main goal of this assignment is to set up an online shop while ensuring the buyer is safe. An online shop for the University of Hull Catering Service was to be built. The dummy system would have a selection of products displayed for its users for them to purchase. After setting up an order, they have an option to pay by cash at collection or to pay through PayPal.

PayPal offers a developer sandbox which can be used to set up dummy buyers and facilitators to test payment transactions. A lot of questions arise about the capabilities of the sandbox, the accounts set up and the weaknesses, security and trustworthiness of the system.

This technical document is divided into two parts. It describes APIs. What information they take, how they work and how the PayPal REST API functions and explains Authenticated Encryption Modes.

The first part is the report which is divided into 3 parts. It will go in depth into APIs, the PayPal Sandbox and the weaknesses in the system. The second part is the short notes which will be depicting the concept of authenticated encryptions modes, their generic composition methods and single-pass authenticated encryption modes. The report then has a conclusion, reference and appendices which contain the pseudo code, the tests made and results, the feedback of the heuristic evaluation and a user manual.

# Report

## API

### *What Is an API*

API stands for Application Programing Interface. It is an intermediate software that allows communication between two applications (Pearlman, 2016). An API is a set of protocols that let the main software applications to be used by another application safely. APIs can be thought of as public methods of an object-oriented program that interact with other elements on the application.

There are different types of API. APIs are divided into four categories: public APIs, partner APIs, internal APIs and composite APIs. Public or Open APIs are APIs that are made publicly available by the software developers. There are to be used free meaning that the software owners give universal access to their customers to integrate this API into their own systems. For example, Facebook’s API allows third-party tools to post on their user’s feed. Partner APIs are not publicly available. Developers will need specific rights or licenses in order to be able to implement them. Internal APIs are also called Private APIs. They are not meant to be used outside of the bound of the company that has developed it. They are integrated among the teams so that they are able to improve their products. The last type of APIs is Composite APIs. They are a combination of APIs that speed up the execution and improve the performance (RapiAPI Staff, 2019).

Certain protocols need to be followed in order to utilise these API types to their full potential. Protocols define a set of rules to be followed by the API calls. It specifies the accepted data types and commands for each API. There are two main protocols: SOAP and REST architectures.

SOAP stands for Simple Object Access Protocol. It is an API for web servers and browsers. This web API standardised how applications use network connections to manage services. Its downsides is that it came with strict rules. Compared to REST APIs, SOAP is more complicated to understand and less flexible.

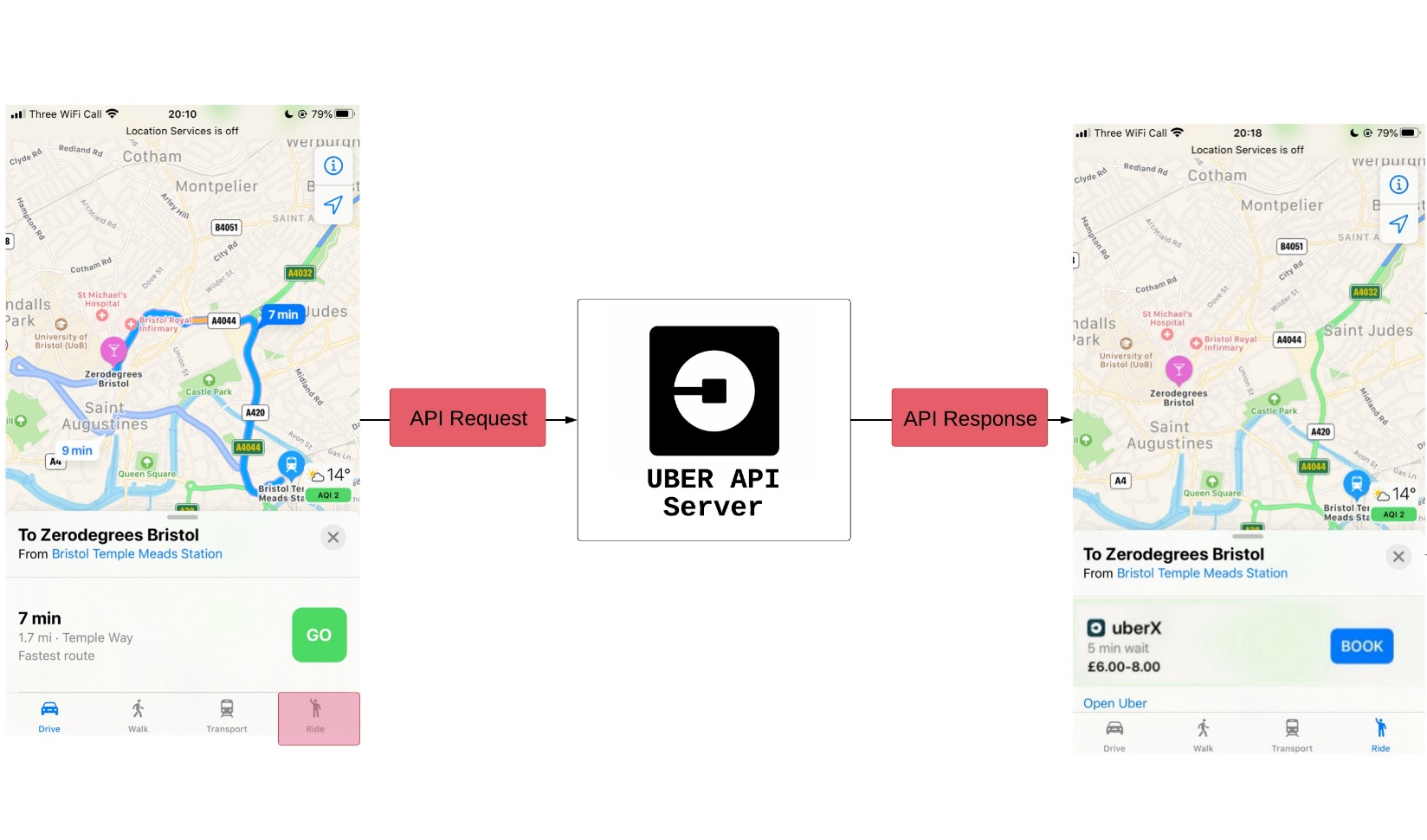
REST stands for Representational State Transfer. It is a web API that takes advantage of existing protocols. It is used on many of the modern web applications such as Netflix, Uber and PayPal. An API is characterised as RESTful when it adheres to the following rules. A REST API is stateless. According to Richardson and Ruby (2007), “statelessness means that every HTTP request happens in complete isolation. When the client makes an HTTP request, it includes all information necessary for the server to fulfill that request. The server never relies on information from previous requests. If that information was important, the client would have sent it again in this request.” A REST API also requires a uniform interface; meaning that the communication need to be done via hypertext transfer protocol using unique resources identifiers, CRUD and JSON conventions. It should be able to cache the response to improve the users’ experience, meaning it would be faster and more efficient to use. This API should also support a layered architecture. Each layer should contribute to hierarchy and allow encapsulation (Castellani and Dorairajan, 2019).

Consider the following scenario: a customer goes for dinner at a restaurant. The customer tells the waiter their order. The latter communicates that order to the kitchen. The kitchen then make the order and gives it to the waiter who delivers it to the customer. The waiter in this case behaves like the API.

### *How Does an API Work*

The first application sends an API request in HTTP to the API server. The server then processes the requests and completes the commands before sending the results from the second application to the first application.

To explain the functionality, let’s take the example of the Uber API used by Apple Maps. Apple Maps offers a ‘Ride’ option that lets the user select a ride services options like Uber. If the user tries to book an Uber through the Apple Maps app, the latter needs to get information from Uber such as the type of ride available, the number of rides available near the user, the price range of the rides and so on. The Uber API server receives those requests and responds by showing the needed info in the Apple Maps application. So, to show the user the price of an uberX ride from the university to the user’s home, Apple would send an HTTP request to Uber with start\_longtitude is the university and end latitude is home (see appendix A for pseudo-code). Uber would then send the information requested and display it in the Apple Maps (Uncubed, 2017).



## Inside the PayPal Sandbox

The PayPal API is a public REST web API. PayPal offers tools that offers resources in order to test and implement this method of payment onto their platform. In order to accomplish that, the users need to set up their development environment and get their personal credentials.

In this section, steps to how to set up an account and how the PayPal API operates will be defined.

### *How to Set Up an Account*

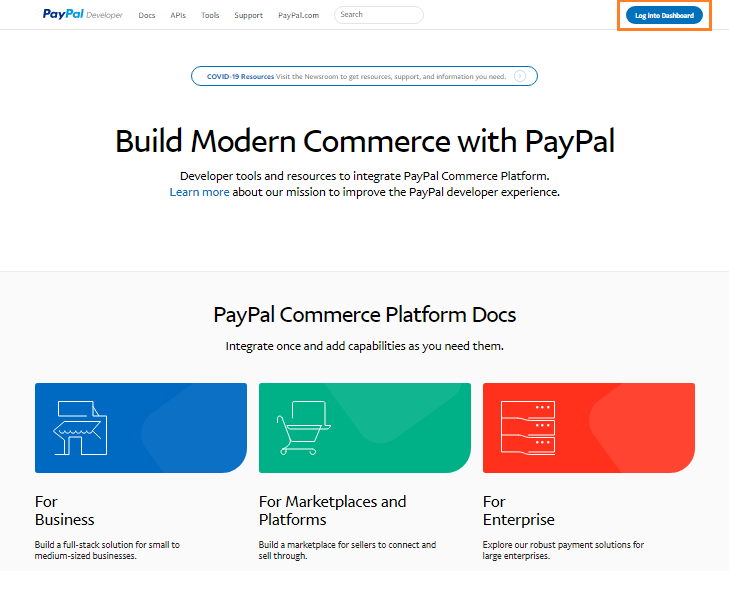
Setting up the PayPal environment occurs in three steps: getting credentials, getting an access token and creating sandbox accounts.

There are three type of accounts that can be acquired to get credentials and create sandbox accounts: a developer account, a personal account or a business account. While all these types can get the necessary credentials, they have different functionalities (PayPal, n.d.).

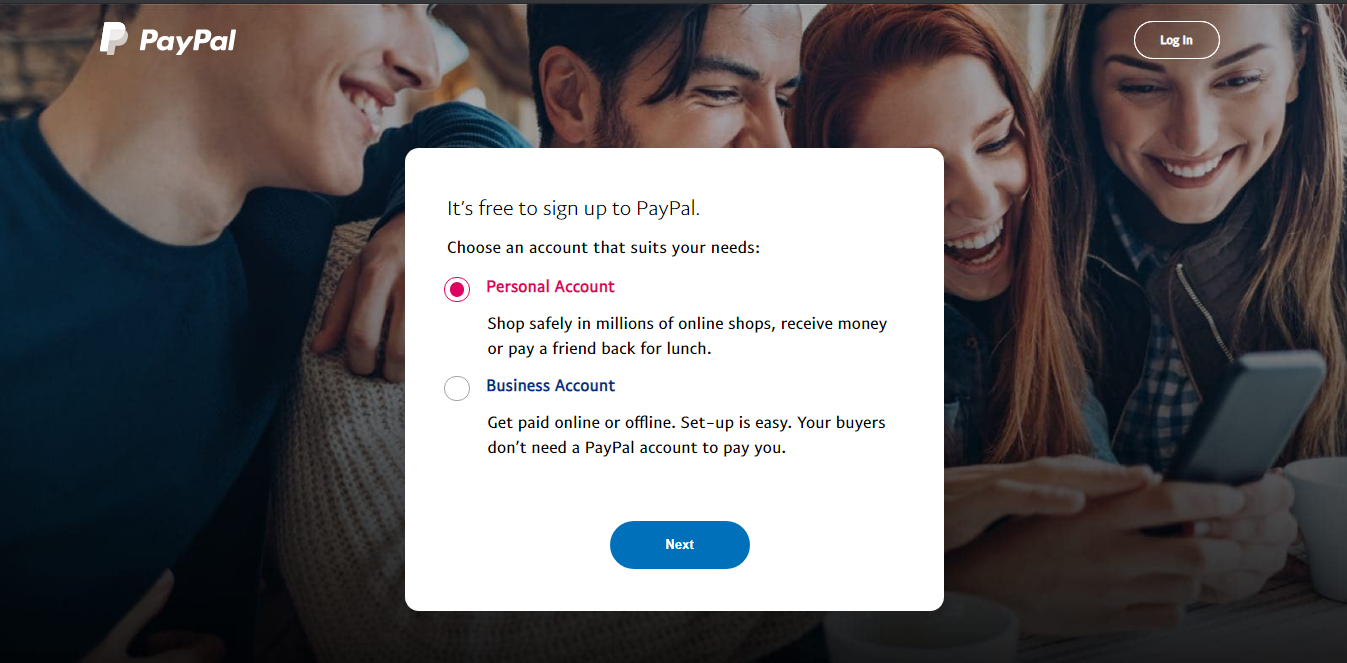
|  |  |  |  |
| --- | --- | --- | --- |
| Capabilities | Developer account | Personal account | Business account |
| Access sandbox | X | X | X |
| Send and receive money |  | X | X |
| Go live |  |  | X |

Sandbox accounts and app credentials were acquired into order to run purchases on the University of Hull Catering Service dummy system. The following are the steps followed so that the payments can be accomplished:

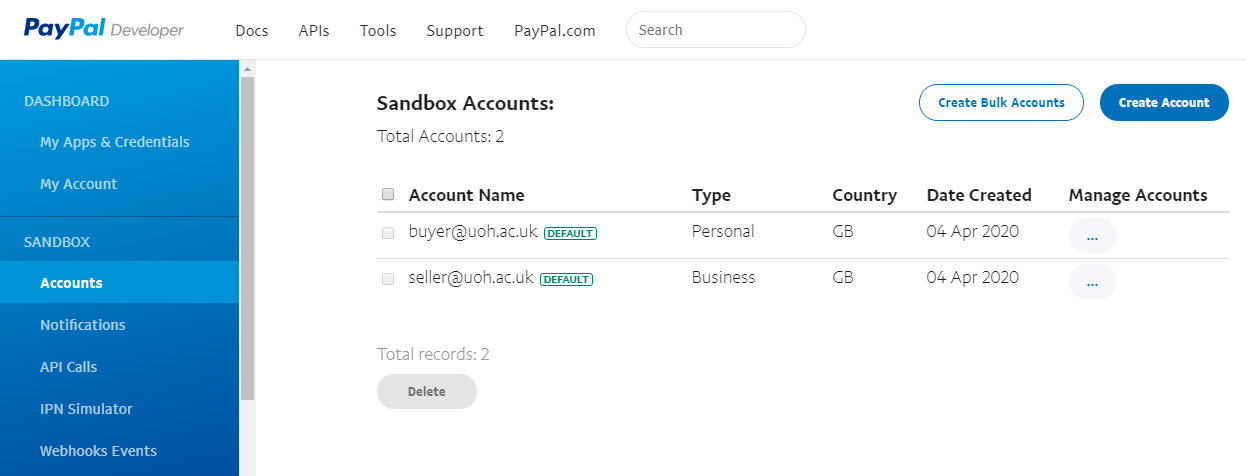
Step 1: Go to PayPal Developer’s website. Click on the ‘Log Into Dashboard’ button.



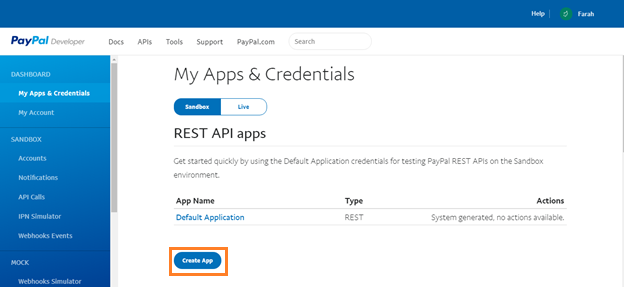
Step 2: You will be redirected to the Log In page. If you do not have an account or wish to have a new account for the developer platform, click Sign Up. You will be directed to the following page. Check for ‘Personal Account’



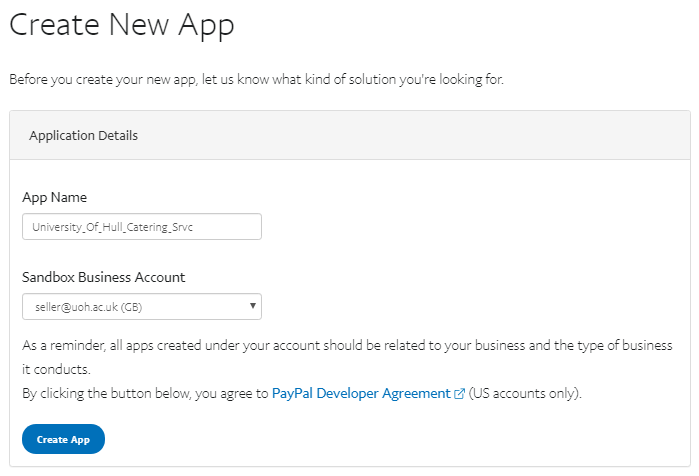
Step 3: After logging, you will be redirected to the Dashboard. The ‘My Apps & Credentials’ page is open by default. Here, you can see the REST API apps and credentials. One app is made by default. For this project, we will be creating new details for this project.

Before doing so, go to ‘Accounts’ under ‘Sandbox’. A personal and business accounts are made by default. The personal account has been renamed to 'buyer@uoh.ac.uk'; the facilitator account has been renamed to 'seller@uoh.ac.uk.'

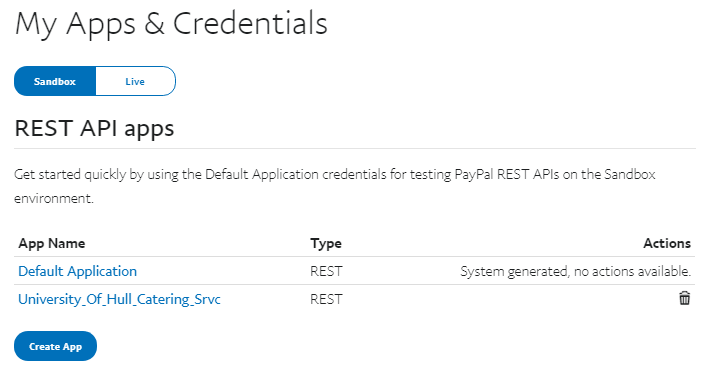
Step 4: Go back to the ‘My Apps & Credentials’ page under ‘Dashboard’ and click ‘Create App’

You will be redirected to the Create App page

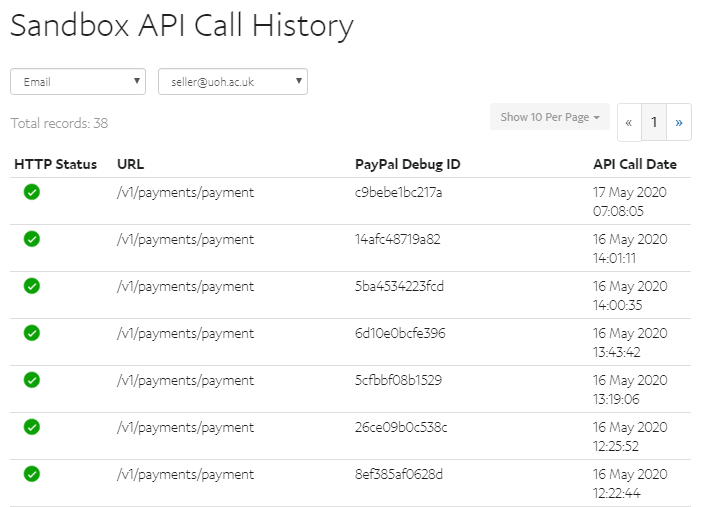
Step 5: You will be redirect to the ‘Create New App’ page where you will be filling out the application details. For this project, the App Name is ‘University\_Of\_Hull\_Catering\_Srvc’ and the Sandbox Business Accounts is set as [seller@uoh.ac.uk](mailto:seller@uoh.ac.uk)



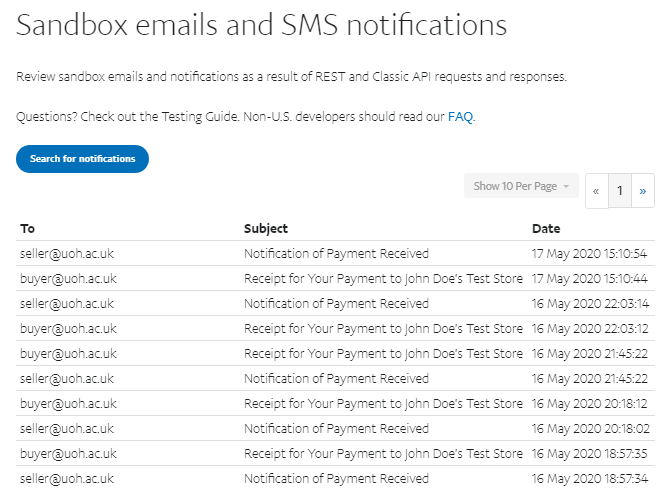
If the creation of the app is successful, you will be redirected to the My Apps & Credentials page where the new application will now show up in the apps table and when clicked, its details are displayed.



In another page, we can see the API calls made when a purchase is done:



After a test payment is done, the PayPal emails with the order confirmation will show up in the ‘Notification’ page.



### *PayPal API*

After completing the development environment creation, the credentials needed to set up payments on your platform are generated and ready to be used as intended. When clicking on the app in ‘My Apps & Credentials’, the OAuth 2.0 credentials for the app will be displayed. The sandbox account is the account to which all the payment is directed. The Client ID and Secret behave as a username and password for the app. The Client ID is used to publicly identify your app. The Client Secret is only known to the application and the authorization server. Its purpose is to protect the communication by granting access tokens to the authorized users only.



To accept payments, the shop must integrate the PayPal mechanisms. Implementing the PayPal button is what made this possible. Once clicked, the PayPal API is called to set up the payment and URL is redirected to the PayPal pop-up where they can complete their purchase.

The PayPal open-source JS code that does the payments is presented below. The catering system request the

## Weakness

### *Dummy Catering System*

* No sessions
* Login, for now hard coded, stored in localstorage - SHA
* If localstorage cleared, order is deleted
* After checkout, if the users goes back to previous page, the order can be processed again
* No way to save the orders previously made
* Order summary doesn’t show in paypal email

### *PayPal Sandbox*

* Glitches in accounts editing
* API time changes

# Short Note

## Authenticated Encryption Modes

## Generic Composition Methods of Authenticated Encryption Modes

## Single-Pass Authenticated Encryption Modes

# Conclusion

# References

Castellani, S. and Dorairajan, A., 2019. *Types of Apis | Learn About the Different Types of APIs in 2020*. [online] API Friends. Available at: <https://apifriends.com/api-creation/different-types-apis/> [Accessed 13 May 2020].

PayPal, n.d. *Get started with PayPal Commerce Platform.* PayPal Developer. [online] Available at: [https://developer.paypal.com/docs/api/overview/#](https://developer.paypal.com/docs/api/overview/) [Accessed 29 April 2020]

Pearlman, S., 2016. *What Are APIs And How Do APIs Work?* [online] MuleSoft Blog. Available at: <https://blogs.mulesoft.com/biz/tech-ramblings-biz/what-are-apis-how-do-apis-work/> [Accessed 13 May 2020].

RapidAPI Staff., 2019. *Types of APIs*. [online] RapidAPI. Available at: <https://rapidapi.com/blog/types-of-apis/> [Accessed 13 May 2020]

Richardson, L. and Ruby, S., 2007. *RESTful Web Services.* Beijing: O'Reilly, p. 86.

Uncubed. 2017. *What Is An API? Learn The Basics In 3 Minutes*. [online] Available at: <https://www.youtube.com/watch?v=Rha1_St_9kw> [Accessed 13 May 2020].

# Appendix

## Appendix A: Pseudo Code

### *Uber API Pseudo-Code*

### *PayPal API Pseudo-Code*

## Appendix B: Tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***PAGE*** | ***#*** | ***TEST CONDUCTED*** | ***EXPECTED RESULT*** | ***ACTUAL RESULT*** |
| Home | 1 | Click login button | Get login pop up |  |
|  | 2 | Click login button with no details | Error shows up |  |
|  | 3 | Click login button with no user ID | Error show up |  |
|  | 4 | Click login button with no password | Error show up |  |
|  | 5 | Click log in with wrong password | Error show up |  |
|  | 6 | Click log in with non-existent account | Error show up |  |
|  | 7 | Login as admin | Redirected to admin |  |
|  | 8 | Login as buyer1 | Redirected to index |  |
|  | 9 | Login as buyer2 | Redirected to index |  |
|  | 10 | Logout button | Redirect to home |  |
|  | 11 | Go to Admin button | Redirect to admin |  |
|  | 12 | Continue Shopping button | Redirect to index |  |
|  |  |  |  |  |
| **Admin** | 13 | Switch between tabs | Show different admin options |  |
|  | 14 | Products – Click on ‘Change Details’ | Go to coming soon |  |
|  | 15 | Stock – Open accordion tab | Show relevant table |  |
|  | 16 | Stock – Open multiple accordion tabs | Show multiple tables |  |
|  | 17 | Stock – Close accordion tab | Close relevant table |  |
|  | 18 | Order – Switch through tabs | Show relevant table |  |
|  | 19 | Go back to login page | Tell the user they’re already logged in |  |
|  | 20 | Logout button on nav bar | Go to home |  |
|  |  |  |  |  |
| **Index** | 21 | Switch between tabs | Show relevant pages |  |
|  | 22 | Click on Home | Go to home |  |
|  | 23 | Click on About | Go to about |  |
|  | 24 | Click on Contact | Go to contact |  |
|  | 25 | Click on cart | Go to cart |  |
|  | 26 | Hover on item | Add to Cart popup |  |
|  | 27 | Add item to cart | Cart icon + 1 |  |
|  | 28 | Add same item to cart | Increment in local storage |  |
|  | 29 | Add multiple items to cart | Add in local storage |  |
| **ShoppingCart** | 30 | Go back to login page | Tell the user they’re already logged in |  |
|  | 31 | Logout button on nav bar | Go to home |  |
| ***PAGE*** | ***#*** | ***TEST CONDUCTED*** | ***EXPECTED RESULT*** | ***ACTUAL RESULT*** |
| **ShoppingCart** | 32 | Tests 22 - 25 | Go to relevant page |  |
|  | 33 | Click on Continue Shopping | Go to index |  |
|  | 34 | Click on Proceed to Payment | Go to payment |  |
|  | 35 | Click on logout | Go to home |  |
|  |  |  |  |  |
| **Payment** | 38 | Tests 22 -25 | Go to relevant page |  |
|  | 39 | Click on Pay by Cash | Go to cashCheckout |  |
|  | 40 | Click on Pay by Paypal | Get Paypal pop up |  |
|  | 41 | Click on logout | Go to home |  |
|  |  |  |  |  |
| **CashCheckout Successful** | 42 | Make a new order | Go to index and clear cart |  |
|  | 43 | Click back button |  |  |
|  | 44 | Click on logout | Go to home |  |
|  |  |  |  |  |
| **PayPal payment pop-up** | 45 | Click on top-right drop down | See product details |  |
|  | 46 | Click proceed to payment | Go to PaypalCheckout |  |
| **PaypalCheckoutSuccessful** | 47 | Make a new order | Go to index and clear cart |  |
|  | 48 | Click back button |  |  |

## Appendix C: Heuristic Evaluation Results