**CPR101**

**Activity 9**

SDLC, Platforms, Development Environments, Version Control

Table of Contents

[1. (5) What is sent via the API from one system to another? What is sent back? 1](#_Toc140426824)

[2. (5) Use api.agify.io to predict the age of a person using your given name and an ISO country code 1](#_Toc140426825)

[3. (5) Use the time zone API request at worldtimeapi.org 1](#_Toc140426826)

[4. (16) Using the above JSON data from worldtimeapi.org, fill in the JSON key / value pairs relating to the descriptions in the table below. 2](#_Toc140426827)

[5. (5) How did you confirm that your location timestamp when converted to data/time was the same as the Location date/time in ISO8601 format in the JSON schema? Show your test and the result. 2](#_Toc140426828)

[SDLC – Software Development Life Cycle 2](#_Toc140426829)

[Software Version 4](#_Toc140426830)

[Bibliography 4](#_Toc140426831)

**Using Mozilla Firefox is strongly recommended for this Activity because it can transform JSON responses into a human readable format.** (Raw Data > "Pretty Print") (#)*is**points for API and Time Zone questions.*

# 1. (5) What is sent via the API from one system to another? What is sent back?

API sends: When one system interacts with another via an API, it sends a request which tells the API what actions to perform. (Emma Burlingame, 2022)

Sent back: After the API receives a request, it sends a JSON response back which includes all the responses to the requests made by the API. (Emma Burlingame, 2022)

# 2. (5) Use api.agify.io to predict the age of a person using your given name and an ISO country code

API URL request: https://api.agify.io/?name=amrik&country\_id=in

JSON response: {"count":34,"name":"amrik","age":59,"country\_id":"IN"}

# 3. (5) Use the time zone API request at worldtimeapi.org

API URL request: http://worldtimeapi.org/api/timezone/asia/tokyo

JSON response: {"abbreviation":"JST","client\_ip":"76.69.130.98","datetime":"2023-07-16T09:48:40.715223+09:00","day\_of\_week":0,"day\_of\_year":197,"dst":false,"dst\_from":null,"dst\_offset":0,"dst\_until":null,"raw\_offset":32400,"timezone":"Asia/Tokyo","unixtime":1689468520,"utc\_datetime":"2023-07-16T00:48:40.715223+00:00","utc\_offset":"+09:00","week\_number":28}

# 

# 4. (16)Using the above JSON data from worldtimeapi.org, fill in the JSON key / value pairs relating to the descriptions in the table below.

|  |  |  |
| --- | --- | --- |
| *See Response Schema* | JSON key | JSON value |
| UTC date/time in ISO8601 format | utc\_datetime | "2023-07-16T00:51:54.687700+00:00" |
| Unix UTC timestamp | unixtime | 1689476478 |
| Unix UTC to location difference | utc\_offset | "+09:00" |
| Location's daylight-saving time difference | dst\_offset | 0 |
| Location date/time in ISO8601 format | offset | "2023-07-16T09:51:54.687700+09:00" |
| How do you calculate the *location's* *timestamp* from the UTC timestamp using JSON keys? | Location Timestamp = UTC Timestamp + Offset for UTC (location timestamp = unixtime + raw\_offset) | *Calculated location timestamp value is:*  1689508878 |

# 5. (5) How did you confirm that your location timestamp when converted to data/time was the same as the Location date/time in ISO8601 format in the JSON schema? Show your test and the result.

To confirm that the location timestamp when converted to data/time was the same as the location date/time in ISO8601 format in the JSON scheme, we can use the formula -> Location timestamp = unix UTC timestamp + UTC time offset \* 3600 (number of seconds in an hour). So, we can get the result as -> Location timestamp = 1689476478 + 9 \* 3600 = 1689508878.‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬‬

# SDLC – Software Development Life Cycle

SDLC – Software Development Life Cycle 54 points = 9 points × 6 items, 75+ words each

**Determine**:

I would like to apply the SDLC to the workshop 8 part 1, I submitted for my IPC144 course through the matrix server. To become comfortable with the assignment, first of all, I read the introductory material provided with the assignment for reference. Then by looking at the output, I made a plan and analyzed how to write the ‘C’ code. The plan consisted of dividing code into functions and then combining them all into a single code.

**Define:**

After analyzing the code, then I looked at the details required in the code. Mainly, the coding assignment required us to break the code into functions and then execute then together to make a program. To fully understand the code, I decided to revisit the concepts of Arrays, Structures, Pointers and Functions, in order to become comfortable with the task. By doing so, I was also able to have a firm grasp of all inputs, processing, and outputs being used in the assignment via the code.

**Design**:

The designing process of the code begin by making a flowchart that consisted of all the inputs, outputs, and processing. Then, I decided to name all the variables, constants and macros before so that while writing code there is a fixed pattern followed for these parameters. Then I wrote all the comments in place where a new block of code/function needed to be started by doing this, I was able to differentiate various code blocks which made my work easier.

**Develop:**

To implement the design into programming source code, I started the process of writing the code by first writing all the headers that needs to be included in the code, then I started the main function that contained all the code and the call to other functions. Afterwards, I wrote all individual functions in different code blocks which performed various functions. Lastly, I called all the functions in the main() process which combined to form a fully functioning program. I also wrote comments to increase human readability of the code and compared the output which the program gave to the output required by the workshop.

**Deliver**:

To deliver and deploy my projects, I first uploaded the .c file containing the code in the ipc144 directory on the matrix server. Then I compiled it using the gcc compiler on the matrix server. Then I ran the command for starting the submission process which then asked for the inputs where required and afterwards told the errors and the line in which they were. Moving further, I fixed the errors shown by the compiler and ran the code which upon successful running gave a success message and submitted the code. Alongside this I also wrote the reflection part which consisted of the theoretical questions related to the assignment and I submitted it alongside.

**D'oh**:

For the ongoing system maintenance and user support, I usually keep track of all the files that I upload on the matrix server. I made appropriate directories and store related files in them. To maintain in-lab work with at-home work, I usually use the GlobalProtect VPN at home and upload the latest file on the matrix server. To make sure that I maintain the first version of the assignment project’s code, I use a version number system that helps me in identifying all the different versions of the code files.

# Software Version

Software Version 5 × 2 points each

A. The name of the software is Microsoft Edge, and the latest version of the software is 114.0.1823.82. (Microsoft, 2023)

B. There are three main components of the version number (114.0.1823.82). The first one is 114 that is the number for a major update which is usually done after a year. The second one is 1823 which is the update for minor changes and usually includes small enhancements and minor feature addition. The third component is 82 which depicts small bug fixes and minor performance enhancements. (Microsoft, 2023)

C. The web browser Microsoft edge is forward compatible in the sense, even after upgrading to a newer version the saved bookmarks, history, login id and customizations of the user remain the same.

D. The web browser Microsoft edge is backward compatible in the sense, that after upgrading to a newer version most of the legacy websites using different kinds of languages and web stacks are able to be viewed by the users without any concerns.

E. <https://learn.microsoft.com/en-us/deployedge/microsoft-edge-relnote-stable-channel>

Release date: July 13, 2023

Microsoft Edge - Fixed various bugs and performance issues. This was a macro update that consisted of more than one bug fixes and performance issues. After this update, some user can see reduction in number of bugs and increase in system performance. (Microsoft, 2023)

# **Bibliography**

Emma Burlingame. (2022, July 29). *How to Use an API: Just the Basics .* Retrieved from technologyadvice.com: https://technologyadvice.com/blog/information-technology/how-to-use-an-api/#key\_components\_of\_apis

Microsoft. (2023, July 13). *Release notes for Microsoft Edge Stable Channel .* Retrieved from learn.microsoft.com: Fixed various bugs and performance issues.