Centennial College – MAPD713 - Enterprise Technologies for Mobile Platforms

**Test 1**

# Introduction

In this test, you are asked to create Node.js application with specified list of features. To complete this task, you can use internet resources as well as lecture notes and code.

This test is an individual assignment. You are not permitted to communicate or exchange code in any manner. Any attempt to do so will result in failing the test by both parties involved.

# Instructions

**Total:** 15 points (15%)

Develop Node.js http server to store and retrieve information using HTTP GET and POST requests and JSON data format.

Upon completion of the test, zip the files and name archive as **MAPD713-T1-YourNameHere.zip** (e.g. **MAPD713-T1-JohnDoe.zip**)and **upload** to Test1\_submission dropbox at e.centennial.

**Version 1** (student number ends with even number: 0, 2, 4, 6, 8):

|  |  |  |
| --- | --- | --- |
|  | **Requirement** | **Points** |
| 1 | The project should have **package.json** file with all required dependencies defined and your name specified in the “author” field | 2 |
| 2 | The server should handle HTTP GET requests at the endpoint:  [http://127.0.0.1:3002/games](http://127.0.0.1:3001/students) | 2 |
| 3 | The server should handle HTTP POST requests with JSON payload at the endpoint:  [http://127.0.0.1:3002/games](http://127.0.0.1:3001/students)  payload example  {"title":"Minecraft", "platform":"XBOX"} | 2 |
| 4 | The server should log the information on start-up:   * server uri (IP address and port) & endpoints information, so that user knows how to access the server   Example:  Server started at [http://127.0.0.1:3002/](http://127.0.0.1:3001/)  Resources:/games | 2 |
| 5 | - The server should store in-memory json payload received in POST request ([http://127.0.0.1:3002/games](http://127.0.0.1:3001/students)) | 3 |
| 6 | - The server should return all stored json objects in the response of GET request ([http://127.0.0.1:3002/games](http://127.0.0.1:3001/students) ) | 2 |
| 7 | Demo your solution with Postman in class | 2 |
|  | **TOTAL** | **15** out of 15 |

**Version 2** (student number ends with uneven number: 1, 3, 5, 7, 9):

|  |  |  |
| --- | --- | --- |
|  | **Requirement** | **Points** |
| 1 | The project should have **package.json** file with all required dependencies defined and your name specified in the “author” field | 2 |
| 2 | The server should handle HTTP GET requests at the endpoint:  [http://127.0.0.1:4002/tablets](http://127.0.0.1:5000/books) | 2 |
| 3 | The server should handle HTTP POST requests with JSON payload at the endpoint:  [http://127.0.0.1:4002/tablets](http://127.0.0.1:5000/books)  payload example  {"model":"Galaxy Tab", "brand":"Samsung"} |  |
| 4 | The server should log the information on start-up:   * server uri (IP address and port) & endpoints information, so that user knows how to access the server   Example:  Server started at [http://127.0.0.1:4002/](http://127.0.0.1:5000/)  Resources:/tablets | 2 |
| 5 | - The server should store in-memory json payload received in POST request ([http://127.0.0.1:4002/tablets](http://127.0.0.1:5000/books)) | 3 |
| 6 | - The server should return all stored json objects in the response of GET request ([http://127.0.0.1:4002/tablets](http://127.0.0.1:5000/books) ) | 2 |
| 7 | Demo your solution with Postman in class | 2 |
|  | **TOTAL** | **15** out of 15 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Grading Schema** |  |  |
|  | **Requirement** | **Your points** | **Max Points** |
| 1 | The project should have **package.json** file with all required dependencies defined & your name specified |  | 2 |
| 2 | The server should handle HTTP GET requests at the endpoint |  | 2 |
| 3 | The server should handle HTTP POST requests with JSON payload at the endpoint: |  | 2 |
| 4 | The server should log the information on start-up: |  | 2 |
| 5 | - The server should store in-memory json payload received in POST request |  | 3 |
| 6 | - The server should display all stored json objects in the browser as a response on GET |  | 2 |
| 7 | Demo |  | 2 |
|  | **TOTAL** |  | **15** out of 15 |