TATA CONSULTANCY SERVICES

Experience certainty.

Quiz Generator

A Node.js training

TATA CONSULTANCY SERVICES

Quiz Generator

Oct 2019
Version 1. 0

Authors:

Rony, De Sousa rony.de.sousa@pwc.com

1. Document revision list

Version No / Date	Change Description	Page nos Reference (Current Version)	Page no Ref (Prev Version)
1.0 Oct 2019	*Initial Version*	All	NA

2. About this document

Purpose

This document serves as a guideline for an evaluation about Node.js. The evaluation is centred on serving files, asynchronous development, file system, express, requests.

Intended Audience

This document is primary intended for trainees focusing on learning Node.js technology.

3. Contents

4. Introduction

Quiz generator is a backend application based on the Node.js framework, focused on providing its users with a tool to perform online evaluations about specific topics.

The application involve the development of the below items:

- One node.js micro service, which will use an input parameter to serve static json objects with questions, related to some specific technology.
- One node.js quiz manager service that will consume the micro service to get the proper json object to do the quiz, and will serve html content in order to complete the application workflow. To achieve that, the quiz manager service should start the user name and password from the UI, validate those details against a DB or even against a hardcoded array, and then will have to capture the quiz area, then processing that information to serve the corresponding quiz options, and returning at the end a summary message.

5. Problem Definition and Requirements

5.1 Application Architecture

The architecture of the application should be designed using as much as possible/needed, asynchronous coding, separated files based on each functionality, separated folders locations for each project (the micro service and the quiz manager service application), and writing logs in every important place.

5.1.1 Client side tools

In order to test the application, whether a web browser or another tool like Postman, Soap UI, or Advanced Rest Client, may be used.

5.2 Application Requirements

5.2.1 General

Create a quiz generator trough a micro service that will serve static json objects associated to some specific area, and trough another application (quiz manager service) that will invoke the micro service. The quiz manager service must return proper json objects in order to provide to the user interface (angular application) the right data to follow the application workflow.

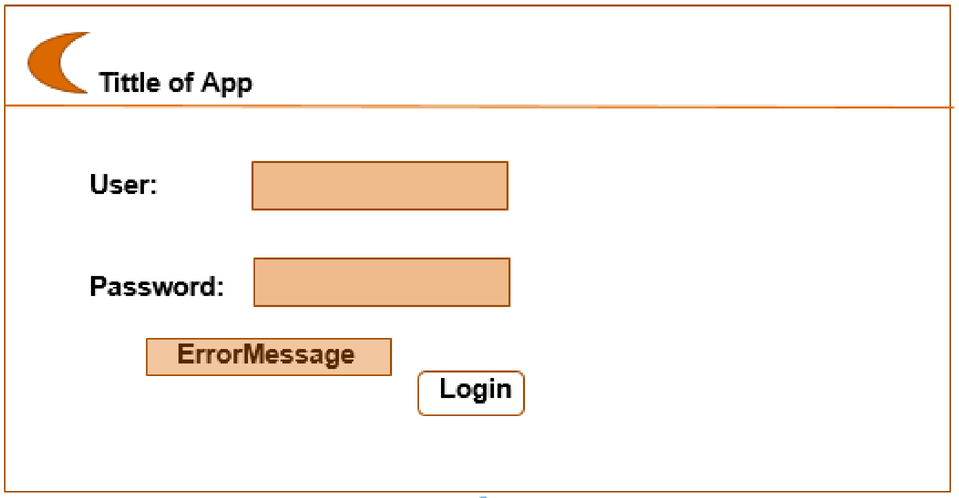
5.2.2 Specifics

- The micro service must have one POST endpoint expecting a json object as parameter.
- The incoming json parameter will indicate to the micro service which specific json object will it serve, based on the required area.
- The micro service must have in their root folder at least three json files to serve them based on the incoming parameter that will require the appropriated json.
- The quiz manager service must have one GET endpoint (getEndPoint1) to capture from the UI the user name and password.
- The quiz manager service must have one additional GET endpoint (getEndPoint2), that will receive from the UI the user name and the area, and then with those parameters will request to the micro service a json object, and must return to the UI the proper json content to start the quiz.
- The quiz manager service must have one last POST endpoint (postEndPoint3) that will return to the UI a “Your quiz has been completed, thanks!” message to the user.
- The quiz manager service should write corresponding (info, warning, debug, error, etc) logs to capture any details that could be helpful from the application support and application development perspective.
- Error handling are required in both applications.

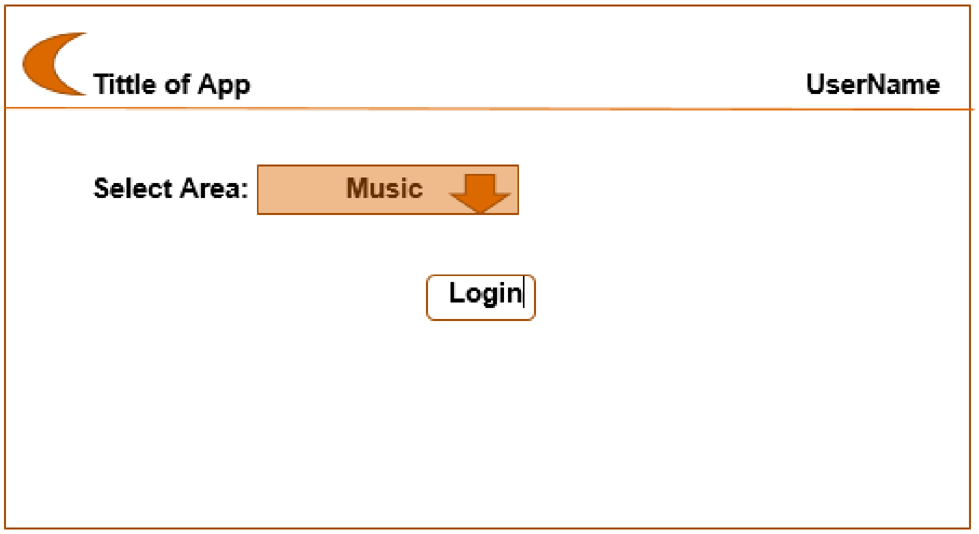
6. Design

6.1 Interface

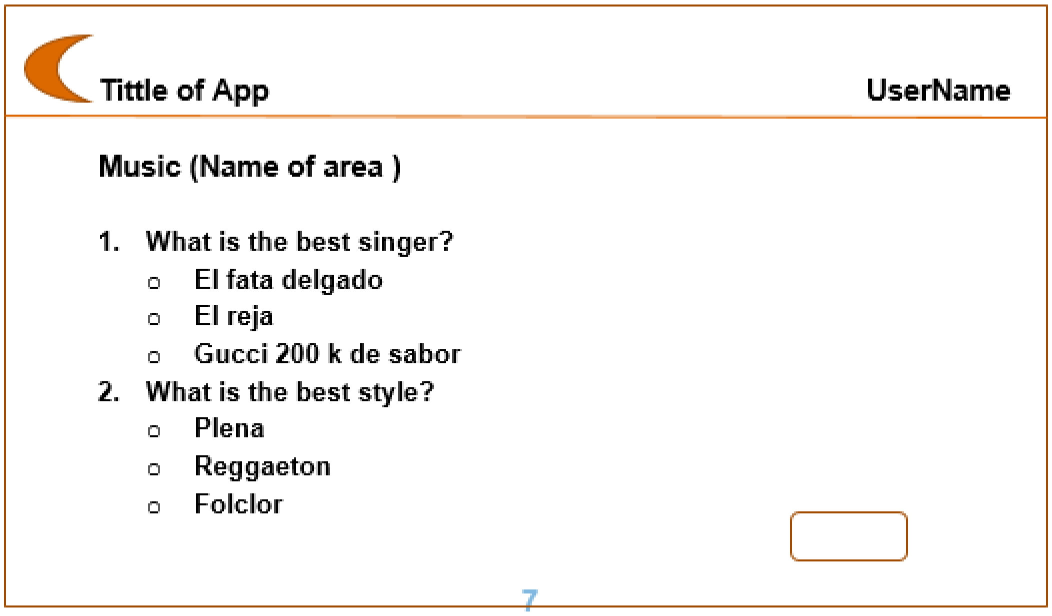
Below are some illustrative images regarding to the application workflow and to the Node.js backend architecture:



Html content provided by the **angular app** (will consume **getEndPoint1**)



Html content provided by the **angular app** (will consume **getEndPoint2**)



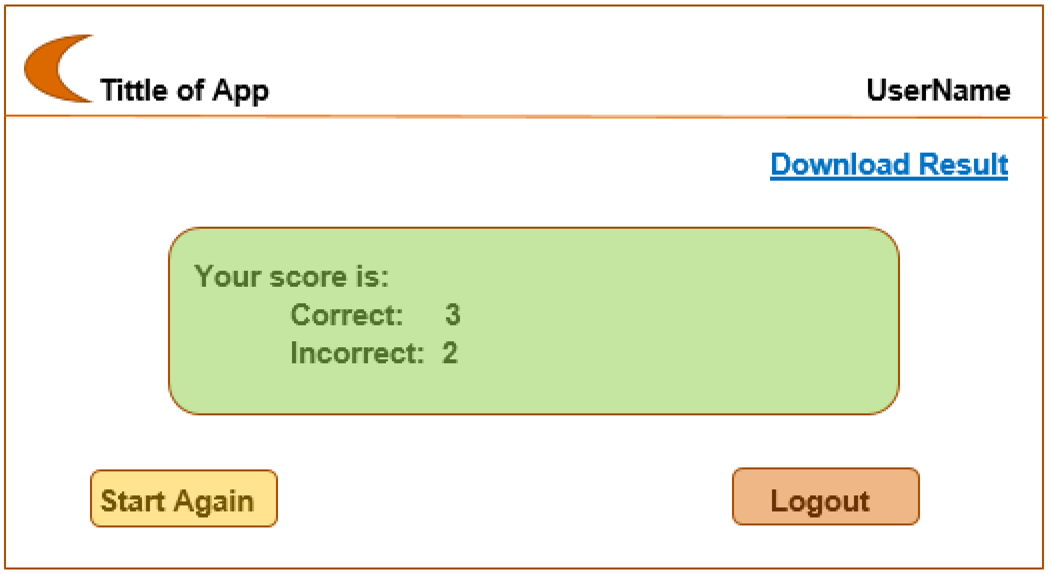
Tittle of App

UserName

Music (Name of area)

1. What is the best singer?
   - El fata delgado
   - El reja
   - Gucci 200 k de sabor
2. What is the best style?
   - Plena
   - Reggaeton
   - Folclor

Html content provided by the **angular app** (will consume **getEndPoint3**)



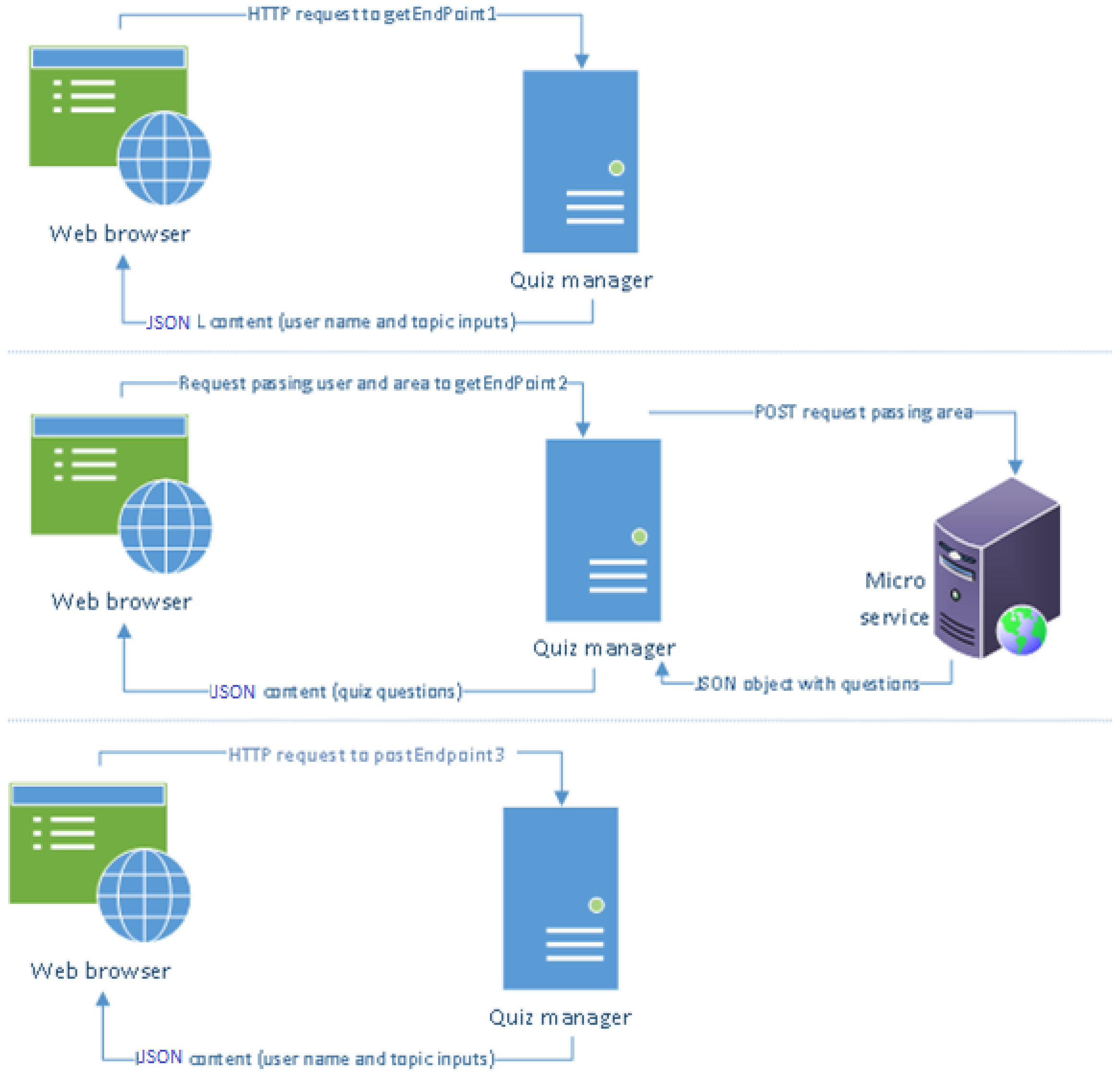
Tittle of App

UserName

Download Result

Your score is:
Correct: 3
Incorrect: 2

Html content provided by the **angular app** (data returned from **getEndPoint3**)


HTTP request to getEndPoint1
Web browser
Quiz manager
JSON L content (user name and topic inputs)
Request passing user and area to getEndPoint2
POST request passing area
Web browser
Quiz manager
Micro service
JSON content (quiz questions)
JSON object with questions
HTTP request to postEndpoint3
Web browser
Quiz manager
JSON content (user name and topic inputs)

Basic quiz workflow

7. JSON Examples

7.1 Node.js json

```
{
    "NodeJs": {
      "q1": {
        "question": "What is the NPM?",
        "options": [
          "Nuget Package Manager",
          "Package manager for the JavaScript programming language",
          "Package manager for Node.js"
        ],
        "answer": "Node Package Management"
      },
      "q2": {
        "question": "Does nodejs run on windows?",
        "options": [
          "Yes",
          "No",
          "Yes, but only nodejs 12.6 and above"
        ],
        "answer": "Yes"
      },
      "q3": {
        "question": "What is ‘callback’ in Node.js?",
        "options": [
          "Just an internal method in any NodeJs application",
          "Function used to deal with multiple requests made to the server",
          "An internal module in NodeJs"
        ],
        "answer": "Function used to deal with multiple requests made to the server"
      },
"q4": {
    "question": "What is express?",
    "options": [
        "An external module",
        "An internal module",
        "A NodeJs I/O command"
    ],
    "answer": "An external module"
},
"q5": {
    "question": "How can you delete a file in Node using fs?",
    "options": [
        "fs.remove(path, callback)",
        "fs.unlink(path, callback)",
        "fs.delete(path, callback)"
    ],
    "answer": "fs.unlink(path, callback)"
},
"q6": {
    "question": "In order to fire events you can do:",
    "options": [
        "emitter.on(eventName, listener)",
        "emitter.listeners(eventName)",
        "emitter.emit(eventName[, ...args])"
    ],
    "answer": "emitter.emit(eventName[, ...args])"
},
"q7": {
    "question": "How to uninstall a dependency using npm?",
    "options": [
        "npm install -u dependency-name",
        "npm uninstall dependency-name",
        "npm unistall -d dependency-name"
    ],
    "answer": "npm uninstall dependency-name"
},
"q8": {
    "question": "What is the Package.json?",
    "options": [
        "File that holds various metadata relevant to the project. This file is used to give information to npm that allows it to identify the project as well as handle the project's dependencies",
        "File automatically generated for any operations where npm modifies either the node_modules tree, or package.json. It describes the exact tree that was generated",
        "File present in the node_modules folder and define all the dependencies for a nodejs app"
    ],
    "answer": "File that holds various metadata relevant to the project. This file is used to give information to npm that allows it to identify the project as well as handle the project's dependencies"
},
"q9": {
    "question": "How can you remove a directory?",
    "options": [
        "fs.rmdir(path, callback)",
        "fs.deletedir(path, callback)",
        "fs.deldirectory(path, callback)"
    ],
    "answer": "fs.rmdir(path, callback)"
},
"q10": {
    "question": "To accommodate the single-threaded event loop, and handle asynchronous events Node.js uses the library?",
    "options": [
        "Libuv",
        "V8Lib",
        "NodejsCore"
    ],
    "answer": "Libuv"
}
```

```
        }
}
```

7.2 Sports json

```
{
        "Sports": {
            "q1": {
                "question": "how many championships Michael Jordan won?",
                "options": [
                    "5",
                    "6",
                    "7"
                ],
                "answer": "6"
            },
            "q2": {
                "question": "The New York Yankees has?",
                "options": [
                    "27 World Series championships",
                    "29 World Series championships",
                    "30 World Series championships"
                ],
                "answer": "27 World Series championships"
            },
            "q3": {
                "question": "Who was Ayrton Senna?",
                "options": [
                    "Brazilian F1 racing driver",
                    "Brazilian motorcycle road racer",
                    "Italian motorcycle road racer"
                ],
                "answer": "Brazilian F1 racing driver"
            },
            "q4": {
                "question": "The team with more championships at the FA Premier League is?",
                "options": [
                    "Rangers Football Club (18)",
                    "Liverpool (18)",
                    "Manchester United (13)"
                ],
                "answer": "Manchester United (13)"
            }
        }
}
```

7.3 Music json

```
{
        "Music": {
            "q1": {
                "question": "The female artist with more Latin Grammys won is?",
                "options": [
                    "Natalia Lafourcade (10)",
                    "Celia Cruz (16)",
                    "Shakira (13)"
                ],
                "answer": "Shakira (13)"
            },
            "q2": {
                "question": "The highest grossing album of 2017 in US?",
                "options": [
                    "Taylor Swift, Reputation",
                    "Ed Sheeran, ÷ (Divide)",
                    "Bruno Mars, 24K Magic"
                ],
                "answer": "Ed Sheeran, ÷ (Divide)"
            },
            "q3": {
                "question": "the highest music viewed video on youtube 2017?",
                "options": [
                    "See You Again - Wiz Khalifa featuring Charlie Puth",
                    "Despacito - Luis Fonsi",
                    "Shape of You - Ed Sheeran"
                ],
                "answer": "Despacito - Luis Fonsi"
            }
        }
}
```

8. Extras

Extras are a required part of the application. Before starting, the trainee should select at least one of the available improvements below, this must to be implemented after finishing the core application.

7.1 Quiz score

As part of the postEndPoint3 the quiz manager service should show to the user the final score, for example: "Your quiz has been completed, thanks! <br /><br />Score: <br />Correct answers: 5 Wrong answers: 3". For that, you will have to return the right json object with accurate data.

7.2 Scheduled job

The quiz manager service should perform an automatic cleanup of the applications logs, moving the files bigger than 1 MB to a backup folder.

7.3 Packages

The logging operations should be implemented through a custom package that will be the responsible of the write logs managements.

7.4 Server cache

The quiz manager service should store at the server cache memory for at least 1 hour every json file in order to reduce the number of requests to the microservice.