**Assignment 9 – Capstone**

**Introduction**

This assignment is a capstone assignment and it is meant to test the knowledge you have acquired throughout the course.

**Business Rules**

* ~~a TEAM can have many players~~
* ~~a TEAM can have one mascot~~

**Instructions**

* ~~Create an ORD based on the provided business rules.~~
* Create a mongoose schema, model, and MongoDB collection that best represents the proposed ORD and business rules.

Team Model

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Schema** |
| name | String | n/a |
| mascot | String | n/a |
| players | Array | playerSchema |

Player Schema

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Schema** |
| firstName | String | n/a |
| lastName | String | n/a |
| salary | Number | n/a |

* Create an API file with four operations: findAllTeams, assignPlayerToTeam, findAllPlayersByTeamId, deleteTeam.

# **findAllTeams**

**Operation: GET**

**Path: /api/teams**

|  |  |  |
| --- | --- | --- |
| **Param Type** | **Field** | **Data Type** |
| n/a | n/a | n/a |

|  |  |
| --- | --- |
| **Response Code** | **Message** |
| 200 | Array of team documents |
| 500 | Server Exception |
| 501 | MongoDB Exception |

**Additional Programming Requirements**

* + Wrap the code in a try/catch block.
  + Describe the operation using the OpenAPI Specification above the request as developer comments.

# **assignPlayerToTeam**

**Operation: POST**

**Path: /api/teams/:id/players**

|  |  |  |
| --- | --- | --- |
| **Param Type** | **Field** | **Data Type** |
| RequestParams | id | String |
| RequestBody | firstName | String |
| RequestBody | lastName | String |
| RequestBody | Salary | String |

|  |  |
| --- | --- |
| **Response Code** | **Message** |
| 200 | Player document |
| 401 | Invalid teamId |
| 500 | Server Exception |
| 501 | MongoDB Exception |

**Additional Programming Requirements**

* + Wrap the code in a try/catch block.
  + Describe the operation using the OpenAPI Specification above the request as developer comments.

# **findAllPlayersByTeamId**

**Operation: GET**

**Path: /api/teams/:id/players**

|  |  |  |
| --- | --- | --- |
| **Param Type** | **Field** | **Data Type** |
| RequestParams | id | String |

|  |  |
| --- | --- |
| **Response Code** | **Message** |
| 200 | Array of player documents |
| 401 | Invalid teamId |
| 500 | Server Exception |
| 501 | MongoDB Exception |

**Additional Programming Requirements**

* + Wrap the code in a try/catch block.
  + Describe the operation using the OpenAPI Specification above the request as developer comments.

# **deleteTeamById**

**Operation: DELETE**

**Path: /api/teams/:id**

|  |  |  |
| --- | --- | --- |
| **Param Type** | **Field** | **Data Type** |
| RequestParams | id | String |

|  |  |
| --- | --- |
| **Response Code** | **Message** |
| 200 | Team document |
| 401 | Invalid teamId |
| 500 | Server Exception |
| 501 | MongoDB Exception |

**Additional Programming Requirements**

* + Wrap the code in a try/catch block.
  + Describe the operation using the OpenAPI Specification above the request as developer comments.
* Test all four API’s using either the generated Swagger documentation at <https://localhost:3000/api-docs> or through SoapUI. Take screenshots of the testing you did from either SoapUI or the Swagger documentation link.
* The screenshots you take of SoapUI and Swagger must be added to your personal portfolio website under the “API Unit Tests Page.” This is a gradable item.
* Deploy the application to Render. This is a gradable item.
  + **Special note**. To deploy your application to **Render**, follow the steps outlined in the **Render Deployment Guide** under the **Weekly Resources** section.
* Add a link to the application in your personal portfolio under the “projects” page. This is a gradable item.
* If you run into issues, refer to the courses GitHub repository and the work you completed in WEB 340.