
NAME :- Aniket Dharma

Employee Code :- 1246

A coach logs into the hudl.com web app and wants to access a schedule page that contains a list of opponents information, including date and time of each game.

TEST CASE 1 :-

A registered and authorized coach wants to create a schedule event for “2020-2021” season.

Steps/Validation:

1) Coach clicks on the “Add an event” button in the current season to create a new schedule entry by entering the details.

2) POST :-

As soon as coach hits the save button, the data is put in the response body of the request and sent to the server in the form of JSON.

```
{  
  
  "gameid: "1234567" ,sqlld: "1234567", date: "2020-01-01T19h00h00", opponent:  
  "TestOpponent", opponentId: "123456", isHome: true, gameType: 0, categories: [  
  ]  
}
```

3) Response :- On success, returns with 200 OK status.

```
{  
  
  Entryid:"1234567","updategame":  
  {  
  
    "gameid: "1234567" ,sqlld: "1234567", date: "2020-01-02T20h00h00", opponent:  
    "TestOpponent", opponentId: "123456", isHome: true, gameType: 0, categories: [  
    ], error: "NULL"  
  }  
}
```

```
}
```

TEST CASE 2 :-

A registered and authorized coach wants to view the schedule events in the season "2020-2021".

Steps/Validation:

1) Coach clicks on the season "2020-2021" to view the schedule entries.

2) **GET :-**

The GET method is used to retrieve data from server at the specific resource.

```
{
```

```
gameid: "1234567" ,sqlId: "1234567", date: "2020-01-01T19h00h00", opponent:
"TestOpponent", opponentId: "123456", isHome: true, gameType: 0, categories: [
]
```

```
}
```

3) **Response :-** On success, returns with 200 OK status.

```
{
```

```
gameid: "1234567" ,sqlId: "1234567", date: "2020-01-02T20h00h00", opponent:
"TestOpponent", opponentId: "123456", isHome: true, gameType: 0, categories: [
], error: "NULL"
```

```
}
```

TEST CASE 3 :-

A registered and authorized coach wants to update the schedule entry in the season "2020-2021".

Steps/Validation:

1) Coach clicks on the particular schedule event he wants to update in the season "2020-2021"

2) PUT :-

The PUT method is used to send data to API to update or create a resource. We are making changes in the date and time of a particular game.

```
{
```

```
gameid: "1234567",sqlld: "1234567", date: "2020-01-02T20h00h00", opponent:
"TestOpponent", opponentId: "123456", isHome: true, gameType: 0, categories: [
]
```

```
}
```

3) On success, returns with 200 OK status.

```
{
```

```
Entryid:"1234567","updategame": {
```

```
gameid: "1234567",sqlld: "1234567", date: "2020-01-02T20h00h00", opponent:
"TestOpponent", opponentId: "123456", isHome: true, gameType: 0, categories: [
], error: "NULL"
```

```
}
```

```
}
```

4) Perform GET method to validate the changes in the schedule entry.

TEST CASE 4:-

A registered and authorized coach wants to delete the schedule entry in the season "2020-2021".

Steps/Validation :-

1) The coach clicks on the particular event and delete the event.

2) DELETE :-

Delete method delete the resources at the specified URL.

{

gameid: "1234567" ,sqlld: "1234567", date: "2020-01-01T19h00h00", opponent: "TestOpponent", opponentId: "123456", isHome: true, gameType: 0, categories: [

]

}

3) Response :-

On success, returns with 200 status code with ok.

TEST CASE 5:-

A registered and authorized coach wants to get the deleted schedule entry.

Steps/Validation:-

1) GET :- GET method is called to get the requested schedule entry

{

```
gameid: "1234567" ,sqlId: "1234567", date: "2020-01-01T19h00h00", opponent:
"TestOpponent", opponentId: "123456", isHome: true, gameType: 0, categories: [
] }
```

2) Response:-

Returns with status code of 404 not found.

TEST CASE 6 :-

Un-authorized coach wants to add a schedule entry

Steps/Validation:-

1) The coach has no access to create a schedule event and he tries to create a schedule event

2) Response :-

Returns with a status code of 401

{

Error :- "Unauthorized"

}

3) Validate that coach needs to be authorized to perform the action.

TEST CASE 7:-

1) A registered and authorized coach is unable to add entry due to server error.

2) Steps/Validation:-

POST :- Post method is called to update the entry into database.

{

```
gameid: "1234567" ,sqlId: "1234567", date: "2020-01-01T19h00h00", opponent:
"TestOpponent", opponentId: "123456", isHome: true, gameType: 0, categories: [
]
}
```

3) Response:-

Returns with status code of 500 server error

```
{
  Error :- "Internal server error "
}
```

TEST CASE 8:-

1) A registered and authorized coach wants to access services which is not accessible due to server limited users accessibility.

Steps/Validation:-

2) PUT :- Put method is called to update entry into database.

```
{
gameid: "1234567" ,sqlId: "1234567", date: "2020-01-02T20h00h00", opponent:
"TestOpponent", opponentId: "123456", isHome: true, gameType: 0, categories: [
]
}
```

3) Response:-

Returns with status code of 403 forbidden

```
{
Error :- "User access is forbidden "
```

```
}
```

TEST CASE 9:-

1) A registered and authorized coach wants to access opponent entry which is not present into database.

2) GET :- The GET method is used to retrieve data from server at the specific resource.

```
{
```

```
gameid: "1234567" ,sqlId: "1234567", date: "2020-01-01T19h00h00", opponent:
"TestOpponent", opponentId: "123456", isHome: true, gameType: 0, categories: [
] }
```

3) Response:-

Returns with status code of 404 Resources not found

```
{
```

Error :- "Not found"

```
}
```
