

Portfolio Assignment: Final Project

CS 493: Cloud Application Development

URL for Application Deployment: <https://creekk-final.uw.r.appspot.com/>

URL for Account Creation / Login: <https://creekk-final.uw.r.appspot.com/>

Fall 2022

Oregon State University

Last Update: 11/29/2022

Change log	2
Data Model	3
Relationships.....	5
Create a Player (Unprotected).....	6
View all Players (Unprotected)	8
View a Specific Player (Unprotected)	10
Edit a Player (Partial Edit) (Unprotected)	11
Edit a Player (COMPLETE Edit) (Unprotected)	13
Delete a Player (Partially Protected)	15
Create a Team (Partially Protected)	17
View a Team (Protected)	19
View a Specific Team (Protected)	21
Edit a Team (Partial Edit) (Protected)	22
Edit a Team (COMPLETE Edit) (Protected).....	24
Delete a Team (Protected).....	26
Add a Player to a Team Roster (Protected)	27
Remove a Player From a Roster (Protected)	29
View All Owners (Unprotected).....	31

Change log

Version	Change	Date
1.0	Initial version.	Nov 29, 2022

Data Model

- For all entities and their properties, see below sections “Owners”, “Teams”, and “Players”
- For description of relationship between all entities, see below section “Relationships”
- User Entity model information:
 - “Owners” represent the User Entity.
 - The unique identifier of the “Owner” is the “username” property.
 - Requests must be accompanied with a valid JSON Web Token to accurately identify a user.
 - The application maps a supplied JWT to a user by associating the JWT with the username, then cross referencing the username with the “Owners” table to determine a unique ID.
 - For relationship between the user entity and non-user entity, reference the below “Relationships” section.

Owners

Property	Type	Required?	Valid Values	Description
Id	Integer	Generated	Automatically Generated By Google	The id of the owner. Datastore automatically generates it. Don't add it yourself as a property of the entity.
username	String	Yes	All String Values. Generally Email Address. Validation Controlled by Auth0	Team owners username.

Teams

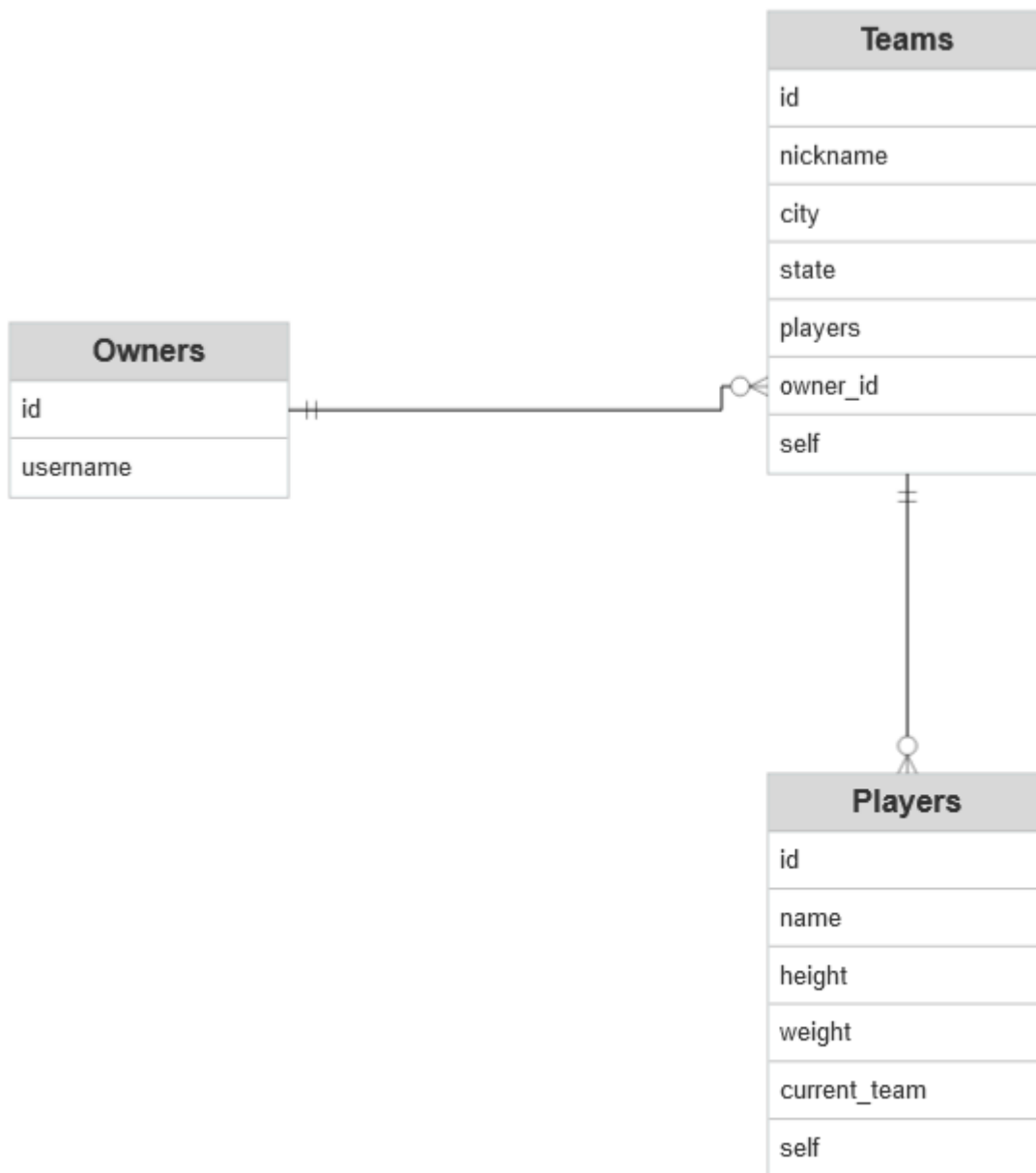
Property	Type	Required?	Valid Values	Description
Id	Integer	Generated	Automatically Generated by Google	The id of the Team. Datastore automatically generates it. Don't add it yourself as a property of the entity.
nickname	String	Yes	All String Values	Team nickname.
city	String	Yes	All String Values	Team home city.
state	String	Yes	All String Values	Team home state.
players	List	Generated	Empty List Generated on instantiation.	List containing each player assigned to the team.

			List is populated with player_id values.	
owner_id	Integer	Generated	Inherited from JWT. Points to owner_id of team owner.	Identifies team owner.
Self	String	Generated	Generated upon instantiation.	Points to the canonical representation of the entity.

Players

Property	Type	Required?	Valid Values	Description
Id	Integer	Generated	Automatically Generated by Google	The id of the player. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	String	Yes	All String Values	Player name.
height	Integer	Yes	All Integer Values	Player height, in inches.
weight	Integer	Yes	All Integer Values	Player weight, in pounds.
current_team	String	Generated	None type on instantiation. Value will be inherited when team is assigned.	Team to which player is currently assigned.
self	String	Generated	Generated upon instantiation.	Points to the canonical representation of the entity.

Relationships



Create a Player (Unprotected)

Creates a Player Entity

POST /players

Request

Path Parameters

None

Request MIME Type

Application/JSON

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Player name.	Yes
height	Player height, in inches.	Yes
weight	Player weight, in pounds.	Yes

Note: Additional attributes “current_team” and “self” will be automatically generated on conception.

Note 2: The “current_team” attribute will be set to “None” on player instantiation.

Note 3: API will not verify for unique “name”.

Request Body Example

```
{
  "name": "Kyle Creek",
  "height": 60,
  "weight": 205
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	

Failure	400 Bad Request	Returned when request body is missing “name”, “height”, or “weight” attribute.
---------	-----------------	--

Response Examples

Success

<p>Status: 201 Created</p> <pre>{ "id": 123, "name": "Kyle Creek", "height": 60, "weight": 205, "current_team": null, "self": [URL Base]/players/[id] }</pre>

Failure

<p>Status: 400 Bad Request</p> <pre>{ "Error": "Missing Attributes" }</pre>

View all Players (Unprotected)

View all players

GET /players

Request

Path Parameters

None

Request MIME Type

Application/JSON

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

```
Status: 200 OK
{
  "players": [
    {
      "height": 60,
      "weight": 205,
      "current_team": null,
      "self": "http://localhost:3000/players/5142858919051264",
      "name": "Kyle Creek 2"
    },
    {
      "height": 60,
      "name": "Kyle Creek",
      "current_team": null,
      "self": "http://localhost:3000/players/5709113312935936",
      "weight": 205
    }
  ]
}
```


}

View a Specific Player (Unprotected)

View information of a specific player id.

GET /players/<player_id>

Request

Path Parameters

Name	Description
player_id	Player ID associated with the requested Player

Request MIME Type

Application/JSON

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	Returned when the provided player id is not associated with any instantiated player.

Response Examples

Success

```
Status: 200 OK
{
  "id": 123,
  "name": "Kyle Creek",
  "height": 60,
  "weight": 205,
  "current_team": null,
  "self": [URL Base]/players/[id]
}
```

Failure

```
Status: 404 Not Found
{
  "Error": "Player not Found"
}
```

Edit a Player (Partial Edit) (Unprotected)

Partially edit aspects of a player instance.

PATCH /players/<player_id>

Request

Path Parameters

Name	Description
player_id	Player ID associated with the requested Player

Request MIME Type

Application/JSON

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Player name.	No
height	Player height, in inches.	No
weight	Player weight, in lbs.	No

Note: Player attribute "current_team", is revised using the "/teams/<team_id>/players/<player_id>" end point.

Note 2: Additional attributes can be sent in the JSON request body, but they will be ignored.

Note 3: Uniqueness will not be verified.

Request Body Example

<pre>{ "name": "Tony Tiger 100", }</pre>
--

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Failure	404 Not Found	Returned when the provided player id is not associated with any instantiated player.
---------	---------------	--

Response Examples

Success

Status: 200 OK <pre>{ "id": 123, "name": "Tony Tiger 100", "height": 60, "weight": 205, "current_team": null, "self": [URL Base]/players/[id] }</pre>

Failure

Status: 404 Not Found <pre>{ "Error": "Player not Found" }</pre>
--

Edit a Player (COMPLETE Edit) (Unprotected)

Edit EVERY Aspect of a player instance.

PUT /players/<player_id>

Request

Path Parameters

Name	Description
player_id	Player ID associated with the requested Player

Request MIME Type

Application/JSON

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Player Name	Yes
height	Player Height	Yes
weight	Player Weight	Yes

Note: Player attribute "current_team", is revised using the "/teams/<team_id>/players/<player_id>" end point.

Note 2: Uniqueness will not be verified.

Request Body Example

<pre>{ "name": "Smoking Joe", "height": 66, "weight": 210, }</pre>
--

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Failure	400 Bad Request	Returned when attributes are missing from request body
Failure	404 Not Found	Returned when the provided player id is not associated with any instantiated player.

Response Examples

Success

<p>Status: 200 OK</p> <pre>{ "id": 123, "name": "Smoking Joe", "height": 66, "weight": 210, "current_team": null, "self": [URL Base]/players/[id] }</pre>

Failure

<p>Status: 400 Bad Request</p> <pre>{ "Error": "Missing Attribute in request body" }</pre>
<p>Status: 404 Not Found</p> <pre>{ "Error": "Player not Found" }</pre>

Delete a Player (Partially Protected)

Deletes a Player instance from the datastore.

```
DELETE /players/<player_id>
```

Note: JSON Web Token **MUST** be present when deleting a player where “current_team” is not set to None. JSON Web Token **MUST** match the JSON Web Token as the Team Owner. Removal can also be performed at the “/teams/<team_id>/players/<player_id>” end point.

Request

Path Parameters

Name	Description
player_id	Player ID associated with the requested Player

Request MIME Type

Application/JSON

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	Returned when the provided player id is not associated with any instantiated player.
Failure	401 Unauthorized	Returned when incorrect or missing JWT are provided. Only applicable when “DELETE” function is performed where player “current_team” attribute is NOT set to None.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found
{
  "Error": "Player not Found"
}
```

Status: 401 Unauthorized

Error Message Dependent on Issue with provided JSON Web Token.

Create a Team (Partially Protected)

Creates a Team Entity

POST /teams

Note: JSON Web Token **MUST** be present when creating a team. Any JSON Web Token, however, can establish a Team.

Request

Path Parameters

None

Request MIME Type

Application/JSON

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
nickname	Team nickname.	Yes
city	Team home city.	Yes
state	Team home state	Yes

Note: Additional attributes “id”, “players”, “owner_id”, and “self” will be automatically generated on conception.

Note 2: “Players” attribute will be an empty list when teams are instantiated.

Note 3: Team cannot be created unless a valid JSON Web Token is provided to the API.

Note 4: API will not verify for unique “city” nor “state”. The attribute “nickname” will be verified for uniqueness.

Request Body Example

```
{
  "nickname": "Seahawks",
  "city": "Seattle",
  "state": "Washington"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	Returned when request body is missing “nickname”, “city”, or “state” attribute.
Failure	401 Unauthorized	Returned when request is sent without a valid JSON Web Token.
Failure	403 Forbidden	Returned when provided “nickname” is not unique.

Response Examples

Success

Status: 201 Created
<pre>{ "id": 123, "nickname": "Seahawks", "city": "Seattle", "state": "Washington" "players": [] "owner_id": 4567 "self": [URL Base]/teams/[id] }</pre>

Failure

Status: 400 Bad Request
<pre>{ "Error": "Missing Attributes" }</pre>
Status: 401 Unauthorized
Error Message Dependent on Issue with provided JSON Web Token.
Status: 403 Forbidden
<pre>{ "Error": "Name is Not Unique" }</pre>

View a Team (Protected)

Views a Team

GET /teams

Request

Path Parameters

None

Request MIME Type

Application/JSON

Request Body

None

Response

Note: API will only return teams that are owned by the “owner_id” associated with the provided JSON Web Token.

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Returned when request is sent without a JSON Web Token

Response Examples

Success

```
Status: 200 OK
{
  "teams": [
    {
      "id": 5640825748848640,
      "nickname": "Raiders",
      "city": "Las Vegas",
      "state": "Nevada",
      "owner_id": 6206954716266496,
      "self": "http://localhost:3000/teams/5640825748848640"
    },
    {
      "id": 5675594515742720,
```

```
    "nickname": "Seahawks",  
    "city": "Seattle",  
    "state": "Washington",  
    "owner_id": 6206954716266496,  
    "self": "http://localhost:3000/teams/5675594515742720"  
  }  
]  
}
```

Failure

Status: 401 Unauthorized
Error Message Dependent on Issue with provided JSON Web Token.

View a Specific Team (Protected)

Views a Team Specific Team

GET /teams/<team_id>

Request

Path Parameters

Name	Description
team_id	Team ID associated with the requested Team

Request MIME Type

Application/JSON

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Returned when request is sent without a JSON Web Token
Failure	404 Not Found	Returned when provided an invalid team_id

Response Examples

Success

Status: 200 OK

Failure

Status: 401 Unauthorized Error Message Dependent on Issue with provided JSON Web Token.
Status 404: Not Found { "Error": "Team not Found" }

Edit a Team (Partial Edit) (Protected)

Partially edit aspects of a team instance.

PATCH /teams/<team_id>

Request

Path Parameters

Name	Description
team_id	Team ID associated with the requested Team

Request MIME Type

Application/JSON

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
nickname	Team nickname	No
city	Team home city	No
state	Team home state	No

Note: The list containing the “players” is revised using the “/teams/<team_id>/players/<player_id>” end point.

Note 2: Additional attributes can be sent in the JSON request body, but they will be ignored.

Note 3: Uniqueness will only be verified for “nickname” attribute.

Note 4: Team can only be revised when provided with valid JSON Web Token

Request Body Example

```
{
  "nickname": "NOT the Rams",
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Failure	404 Not Found	Returned when provided an invalid team_id
Failure	401 Unauthorized	Returned when owner_id associated with the provided JWT does not match the owner_id associated with the team instance.
Failure	403 Forbidden	Returned attempting to revise team "nickname" to "nickname" that already exists.

Response Examples

Success

<p>Status: 200 OK</p> <pre>{ "id": 123, "nickname": "NOT the Rams", "city": "Seattle", "state": "Washington" "players": [] "owner_id": 4567 "self": [URL Base]/teams/[id] }</pre>

Failure

<p>Status: 401 Unauthorized</p> <p>Error Message Dependent on Issue with provided JSON Web Token.</p>
<p>Status: 404 Not Found</p> <pre>{ "Error": "Team not Found" }</pre>
<p>Status 403 Forbidden</p> <pre>{ "Error": "Name Not Unique" }</pre>

Edit a Team (COMPLETE Edit) (Protected)

Edit every aspect of a team instance.

PUT /teams/<team_id>

Request

Path Parameters

Name	Description
team_id	Team ID associated with the requested Team

Request MIME Type

Application/JSON

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
nickname	Team nickname	Yes
city	Team home city	Yes
state	Team home state	Yes

Note: The list containing the “players” is revised using the “/teams/<team_id>/players/<player_id>” end point.

Note 2: Team can only be revised when provided with valid JSON Web Token

Note 3: Uniqueness will only be verified for “nickname” attribute.

Request Body Example

```
{
  "nickname": "NOT the Rams",
  "city": "Charleston",
  "state": "Confusion"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Failure	404 Not Found	Returned when provided an invalid team_id
Failure	401 Unauthorized	Returned when owner_id associated with the provided JWT does not match the owner_id associated with the team instance.
Failure	400 Bad Request	Returned when request body is missing “nickname”, “city”, or “state” attribute.
Failure	403 Forbidden	Returned attempting to revise team “nickname” to “nickname” that already exists.

Response Examples

Success

Status: 200 OK <pre>{ "id": 123, "nickname": "NOT the Rams", "city": "Charleston", "state": "Confusion" "players": [] "owner_id": 4567 "self": [URL Base]/teams/[id] }</pre>

Failure

Status: 401 Unauthorized Error Message Dependent on Issue with provided JSON Web Token.
Status: 404 Not Found <pre>{ "Error": "Team not Found" }</pre>
Status: 400 Bad Request <pre>{ "Error": "Missing Attributes" }</pre>
Status: 403 Forbidden <pre>{ "Error": "Name Not Unique" }</pre>

Delete a Team (Protected)

Removes a team from the datastore

Note: When a team is deleted where players are on a “roster”, each player will have their “current_team” attribute revised to “None”

```
DELETE /teams/<team_id>
```

Request

Path Parameters

Name	Description
team_id	Team ID associated with the requested Team

Request MIME Type

Application/JSON

Request Body

None

Response

Response Body Format

None

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Returned when owner_id associated with the provided JWT does not match the owner_id associated with the team instance.
Failure	404 Not Found	Returned when provided an invalid team_id

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized  
Error Message Dependent on Issue with provided JSON Web Token.
```

```
Status: 404 Not Found  
{  
  "Error": "Team not Found"  
}
```

Add a Player to a Team Roster (Protected)

Adds a Player to a team Roster

PATCH /teams/<team_id>/players/<player_id>
--

Request

Path Parameters

Name	Description
team_id	Team ID associated with the requested Team
player_id	Player ID associated with the requested Player

Request MIME Type

Application/JSON

Request Body

None

Response

Note: Adding a player to a roster will update the players “current_team” attribute.

Note 2: JWT must correspond to the owner_id associated with the team to perform add.

Note 3: Players can only be added to a team while their “current_team” attribute is set to “None”.

Response Body Format

None

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Returned when owner_id associated with the provided JWT does not match the owner_id associated with the team instance.
Failure	404 Not Found	Returned when provided an invalid team_id or invalid player_id
Failure	406 Not Acceptable	Returned when the Player is already assigned to a team.

Response Examples

Success

Status: 200 OK

Failure

Status: 401 Unauthorized Error Message Dependent on Issue with provided JSON Web Token.
--

Status: 404 Not Found { "Error": "Player/Team not Found" }
Status: 406 Not Acceptable { "Error": "Player is already on a team" }

Remove a Player From a Roster (Protected)

Removes a Player to a team Roster

```
DELETE /teams/<team_id>/players/<player_id>
```

Request

Path Parameters

Name	Description
team_id	Team ID associated with the requested Team
player_id	Player ID associated with the requested Player

Request MIME Type

Application/JSON

Request Body

None

Response

Note: Removing a player to a roster will update the players “current_team” attribute.

Note 2: JWT must correspond to the owner_id associated with the team to perform removal.

Note 3: Players can only be removed from a team while their “current_team” attribute is set to “None”.

Response Body Format

None

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Returned when owner_id associated with the provided JWT does not match the owner_id associated with the team instance.
Failure	404 Not Found	Returned when provided an invalid team_id or invalid player_id

Response Examples

Success

```
Status: 200 OK
```

Failure

```
Status: 401 Unauthorized  
Error Message Dependent on Issue with provided JSON Web Token.
```

Status: 404 Not Found

```
{  
  "Error": "Player/Team not Found"  
}
```

View All Owners (Unprotected)

View all the owners

GET /owners

Request

Path Parameters

None

Request MIME Type

Application/JSON

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK

```
{
  "owners": [
    {
      "username": "user2@test.com"
    },
    {
      "username": "user4@test.com"
    },
    {
      "username": "user3@test.com"
    },
    {
      "username": "user1@test.com"
    }
  ]
}
```

