Portfolio Project: API Spec

CS 493: Cloud Application Development

Fall 2022

Sam Ellison

Link to API: https://portfolio-ellisos2-370318.uk.r.appspot.com

---

Index

Data Model

The app authenticates users as ‘beekeepers’, who can, after authentication, add the protected entity hives to the database. These ‘beekeeper’ users will have a userId stored when they are created. This userId uniquely identifies each user, but is not used in any API calls.

The users entity is related to the protected hives entity. Additionally, there is an unprotected entity, queens, which is related to the hives entity. Hives are automatically assigned to the user that creates them, while queens can be created by anyone. However, queens can only be assigned to hives by the user who owns the hive. Hives identify a queen associated with them with a ‘queen’ field, which is null on creation. Similarly, queens have a ‘hive’ field that is filled when that queen is successfully added to a hive.

The API uses Google’s OAuth API to generate a JWT for a new user. For each request to a protected endpoint (/hives), the JWT must be included in the Authorization header of the request. Each protected endpoint will first authenticate the JWT. Afterwards, the ‘sub’ field of the JWT is extracted and compared to the userId corresponding to the entity associated with the request. For example, if a PATCH request is sent to /hives/:hiveId, the userId of the user who created the hive with hiveId will be checked against the ‘sub’ field of the JWT sent in the request. If there is not a match, the request will be rejected.

Users

Property	**Data Type**	**Notes**
id	Integer	The ID of the user. Automatically generated by Datastore.
userId	String	A separate unique identifier for users. This is equivalent to the sub field of the JWT generated for this user.
firstName	String	The user’s first name. Taken from the user’s Google profile.
lastName	String	The user’s last name. Taken from the user’s Google profile.
isBeekeeper	Boolean	Set to true for all new users. A flag declaring this user is a registered beekeeper.

Hives (Protected)

Property	Data Type	Notes
id	Integer	The ID of the hive. Automatically generated by Datastore.
hiveName	String	Name given to this hive. **Constraints:** Must consist of only valid alphanumeric characters and spaces.
structureType	String	The structure type of this beehive. **Constraints:** Must consist of only valid alphanumeric characters and spaces.
colonySize	Integer	The approximate number of bees in this hive. Type is not verified.
beekeeper	String	The userId for this hive's beekeeper. Automatically assigned as the user that creates this hive.
queen	Object	The object representing this hive's current queen. Can only be changed by the beekeeper. Initialized to null.

Queens (Unprotected)

Property	Data Type	Notes
id	Integer	The ID of the queen. Automatically generated by Datastore.
name	String	Name given to this queen. **Constraints:** Must consist of only valid alphanumeric characters and spaces.
species	String	The species of this bee. **Constraints:** Must consist of only valid alphanumeric characters and spaces.
age	Integer	The approximate age of the queen bee, in months. Type is not verified.
hive	Object	The object representing this queen's current hive. Can only be changed by the beekeeper. Initialized to null.

Get all users (Unprotected)

List all users currently stored. This endpoint is not protected.

```
GET /users
```

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

```
Status: 200 OK
{
    "users": [
        {
            "lastName": "Ellison",
            "userId": "104955094524655810629",
            "isBeekeeper": true,
            "firstName": "Samuel",
            "id": "5738709764800512"
         }
    ]
}
```

Create a hive (Protected)

Allows you to create a new hive entity. Requires the JWT associated with a user to be sent in the authorization header for authentication.

```
POST /hives
```

Request

Headers

Header	Type	Contents
Authorization	Bearer Token	User’s JWT

Path Parameters

None

Request Body

Required

Request Body Format

JSON (application/json)

Request JSON Attributes

Name	Description	Required?
hiveName	Name given to this hive. Must consist of only valid alphanumeric characters and spaces.	Yes
structureType	The structure type of this beehive. Must consist of only valid alphanumeric characters and spaces.	Yes
colonySize	The approximate number of bees in this hive.	Yes

Request Body Example

```
{
   "hiveName": "HV463",
   "structureType": "Clay Hive",
   "colonySize": 24000
}
```

Response

Response Body Format

JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, if non-alphanumeric characters are used in the hiveName or structureType fields, the hive is not created and status code 400 is returned.
Failure	401 Unauthorized	If the JWT is missing or invalid, the hive is not created and status code 401 is returned.
Failure	406 Not Acceptable	Returned if an unsupported MIME type is requested. The server will allow for only application/json in the response for a POST request.
Failure	415 Unsupported Media Type	Returned if the client request data is not sent as application/json.
Failure	500 Internal Server Error	Returned in the event of an unknown server error.

Response Examples

Success

```
Status: 201 Created
{
   "id": "5647975057457152",
   "hiveName": "HV463",
   "structureType": "Clay Hive",
   "colonySize": 24000,
   "queen": null,
   "beekeeper": "104955094524655810629",
   "self": "http://my.url/hives/5647975057457152"
}
```

Failure

```
Status: 400 Bad Request

{
        "Error":  "The request object is missing at least one of the required attributes"
}
```

```
Status: 400 Bad Request

{
        "Error":  “hiveName and structureType must include only alphanumeric
        characters”
}
```

```
Status: 401 Unauthorized

{
        "Error": “Missing or invalid JWT”
}
```

```
Status: 406 Not Acceptable

{
        "Error": "Unsupported MIME type requested - only application/json supported"
}
```

```
Status: 415 Unsupported Media Type

{
        "Error": "Unsupported MIME type received - server can only accept
        application/json"
}
```

Get a hive (Protected)

Allows you to get an existing hive entity. Requires the JWT associated with the user who created the hive to be sent in the authorization header for authentication.

```
GET /hives/:hive_id
```

Request

Headers

Header	Type	Contents
Authorization	Bearer Token	User’s JWT

Path Parameters

Name	Description
hive_id	ID of the hive

Request Body

None

Response

Response Body Format

JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If the JWT is missing or invalid, status code 401 is returned.
Failure	403 Forbidden	The hive is owned by a different user.
Failure	404 Not Found	No hive with this hive_id exists
Failure	406 Not Acceptable	Returned if an unsupported MIME type is requested. The server will allow for only application/json in the response for a GET request.

Response Examples

Success

```
Status: 200 OK [Content-Type: application/json]

{
   "structureType": "Clay Hive",
   "beekeeper": "104955094524655810629",
   "hiveName": "HV463",
   "queen": null,
   "colonySize": 24000,
   "id": "5647975057457152",
   "self": "http://my.url/hives/5647975057457152"
}
```

Failure

Status: 401 Unauthorized

```
{
        "Error": “Missing or invalid JWT”
}
```

Status: 403 Forbidden

```
{
        "Error": “Hive has a different owner"
}
```

Status: 404 Not Found

```
{
        "Error":  "Hive and/or queen not found”
}
```

Status: 406 Not Acceptable

```
{
        "Error": "Unsupported MIME type requested - only application/json supported"
}
```

List all hives for user (Protected)

List all the hive entities for the user whose JWT is sent in the headers for this request. Results will be paginated to show 5 entities at a time.

```
GET /hives
```

Request

Headers

Header	Type	Contents
Authorization	Bearer Token	User’s JWT

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If the JWT is missing or invalid, status code 401 is returned.
Failure	406 Not Acceptable	Returned if an unsupported MIME type is requested. The server will allow for only application/json in the response for a GET request.

Response Examples

Success

```
Status: 200 OK
{
    "total": 16,
    "hives": [
        {
            "beekeeper": "1049550945246558106299",
            "queen": null,
            "colonySize": 3900,
            "structureType": "Clay Hive",
            "hiveName": "HV98",
            "id": "4787246431469568"
        },
        {
            "beekeeper": "1049550945246558106299",
            "queen": null,
            "hiveName": "HV464",
            "colonySize": 1400,
            "structureType": "Clay Hive",
            "id": "4858468934615040"
        },
        {
            "beekeeper": "1049550945246558106299",
            "queen": null,
            "colonySize": 6700,
            "structureType": "Clay Hive",
            "hiveName": "HV465",
            "id": "5075408403824640"
        },
        {
            "beekeeper": "1049550945246558106299",
            "queen": null,
            "colonySize": 4500,
            "hiveName": "HV762",
            "structureType": "Top bar Hive",
            "id": "5083538508480512"
        },
        {
            "queen": {
                "id": 5149887163269120,
                "self": "http://my.url/queens/5149887163269120"
            },
            "beekeeper": "1049550945246558106299",
            "structureType": "Clay Hive",
            "hiveName": "HV463",
            "colonySize": 24000,
            "id": "5108686347304960"
        }
    ],
    "next": "http://my.url/hives?cursor=CjASKmoUZH5wyAgNiryokJDBgAIAA="
}
```

Failure

```
Status: 401 Unauthorized

{
        "Error": “Missing or invalid JWT”
}
```

```
Status: 406 Not Acceptable

{
        "Error": "Unsupported MIME type requested - only application/json supported"
}
```

Delete a hive (Protected)

Allows you to delete a hive. Hives can only be deleted by the authenticated user who created them.

```
DELETE /hives/:hive_id
```

Request

Headers

Header	Type	Contents
Authorization	Bearer Token	User’s JWT

Path Parameters

Name	Description
hive_id	ID of the hive

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	If the JWT is missing or invalid, status code 401 is returned.
Failure	403 Forbidden	The hive is owned by a different user.
Failure	404 Not Found	No hive with this hive_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized

{
        "Error": “Missing or invalid JWT”
}
```

```
Status: 403 Forbidden

{
        "Error": “Hive has a different owner"
}
```

```
Status: 404 Not Found

{
        "Error": “Hive and/or queen not found”
}
```

Update a hive (Protected)

Allows you to update any or all attributes of an existing hive entity. Hives can only be updated by the user who created them, and their generated JWT must be present in the authorization header of the request.

```
PATCH /hives/:hive_id
```

Request

Headers

Header	Type	Contents
Authorization	Bearer Token	User’s JWT

Path Parameters

Name	Description
hive_id	ID of the hive

Request Body

Required

Request Body Format

JSON (application/json)

Request JSON Attributes

Name	Description	Required?
hiveName	Name given to this hive.	No
structureType	The structure type of this beehive.	No
colonySize	The approximate number of bees in this hive.	No

Request Body Example

```
{
    "structureType": "AZ Hive"
}
```

Response

Response Body Format

JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If non-alphanumeric characters are used in the hiveName or structureType fields, the hive is not created and status code 400 is returned.
Failure	401 Unauthorized	If the JWT is missing or invalid, the hive is not created and status code 401 is returned.
Failure	404 Not Found	No hive with this hive_id exists
Failure	406 Not Acceptable	Returned if an unsupported MIME type is requested. The server will allow for only application/json in the response for a PATCH request.
Failure	415 Unsupported Media Type	Returned if the client request data is not sent as application/json.

Response Examples

Success

```
Status: 200 OK

Location: http://my.url/hives/5647975057457152

{
    "colonySize": 24000,
    "structureType": "AZ Hive",
    "beekeeper": "104955094524655810629",
    "hiveName": "HV463",
    "queen": null,
    "id": "5647975057457152",
    "self": "http://my.url/hives/5647975057457152"
}
```

Failure

```
Status: 400 Bad Request

{
        "Error":  “hiveName and structureType must include only alphanumeric
        characters”
}
```

```
Status: 401 Unauthorized

{
        "Error": “Missing or invalid JWT”
}
```

```
Status: 404 Not Found

{
        "Error": “Hive and/or queen not found”
}
```

```
Status: 406 Not Acceptable

{
        "Error": "Unsupported MIME type requested - only application/json supported"
}
```

```
Status: 415 Unsupported Media Type

{
        "Error": "Unsupported MIME type received - server can only accept
        application/json"
}
```

Replace a hive (Protected)

Allows you to replace all base attributes of an existing hive entity. Hives can only be updated by the user who created them, and their generated JWT must be present in the authorization header of the request.

```
PUT /hives/:hive_id
```

Request

Headers

Header	Type	Contents
Authorization	Bearer Token	User’s JWT

Path Parameters

Name	Description
hive_id	ID of the hive

Request Body

Required

Request Body Format

JSON (application/json)

Request JSON Attributes

Name	Description	Required?
hiveName	Name given to this hive.	Yes
structureType	The structure type of this beehive.	Yes
colonySize	The approximate number of bees in this hive.	Yes

Request Body Example

```
{
    "hiveName": "HV762",
    "structureType": "Top bar Hive",
    "colonySize": 4500
}
```

Response

Response Body Format

JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	A link to the updated hive page will be placed in the response Location header.
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, if non-alphanumeric characters are used in the hiveName or structureType fields, the hive is not created and status code 400 is returned.
Failure	401 Unauthorized	If the JWT is missing or invalid, the hive is not created and status code 401 is returned.
Failure	404 Not Found	No hive with this hive_id exists
Failure	406 Not Acceptable	Returned if an unsupported MIME type is requested. The server will allow for only application/json in the response for a PUT request.
Failure	415 Unsupported Media Type	Returned if the client request data is not sent as application/json.

Response Examples

Success

```
Status: 303 See Other

Location: http://my.url/hives/5647975057457152

{
   "id": 5647975057457152,
   "hiveName": "HV762",
   "structureType": "Top bar Hive",
   "colonySize": 4500,
   "beekeeper": "104955094524655810629",
   "queen": null,
   "self": "http://my.url/hives/5647975057457152"
}
```

Failure

```
Status: 400 Bad Request

{
         "Error":  “hiveName and structureType must include only alphanumeric
         characters”
}
```

```
Status: 401 Unauthorized

{
         "Error": “Missing or invalid JWT”
}
```

```
Status: 404 Not Found

{
         "Error": “Hive and/or queen not found”
}
```

```
Status: 406 Not Acceptable

{
         "Error": "Unsupported MIME type requested - only application/json supported"
}
```

```
Status: 415 Unsupported Media Type

{
         "Error": "Unsupported MIME type received - server can only accept
         application/json"
}
```

Create a queen (Unprotected)

Allows you to create a new queen entity. This endpoint is not protected. However, assigning a queen to a hive can only be done by authenticated users.

```
POST /queens
```

Request

Headers

None

Path Parameters

None

Request Body

Required

Request Body Format

JSON (application/json)

Request JSON Attributes

Name	Description	Required?
name	Name given to this queen. Must consist of only valid alphanumeric characters and spaces.	Yes
species	The species of this bee. Must consist of only valid alphanumeric characters and spaces.	Yes
age	The approximate age of the queen bee, in months. Type is not verified.	Yes

Request Body Example

```
{
   "name": "Eugenie",
   "species": "Western Honey Bee",
   "age": 4
}
```

Response

Response Body Format

JSON (application/json)

Response Statuses

Outcome	**Status Code**	**Notes**
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, if non-alphanumeric characters are used in the name or species fields, the queen is not created and status code 400 is returned.
Failure	406 Not Acceptable	Returned if an unsupported MIME type is requested. The server will allow for only application/json in the response for a POST request.
Failure	415 Unsupported Media Type	Returned if the client request data is not sent as application/json.
Failure	500 Internal Server Error	Returned in the event of an unknown server error.

Response Examples

Success

```
Status: 201 Created


{
   "id": "5669152601669632",
   "name": "Eugenie",
   "species": "Western Honey Bee",
   "age": 4,
   "hive": null,
   "self": "http://my.url/queens/5669152601669632"
}
```

Failure

```
Status: 400 Bad Request

{
        "Error":  "The request object is missing at least one of the required attributes"
}
```

```
Status: 400 Bad Request

{
        "Error":  “Name and species must include only alphanumeric characters”
}
```

```
Status: 406 Not Acceptable

{
        "Error": "Unsupported MIME type requested - only application/json supported"
}
```

```
Status: 415 Unsupported Media Type

{
        "Error": "Unsupported MIME type received - server can only accept
        application/json"
}
```

Get a queen (Unprotected)

Allows you to get an existing queen entity. This endpoint is not protected. However, assigning a queen to a hive can only be done by authenticated users.

```
GET /queens/:queen_id
```

Request

Headers

None

Path Parameters

Name	Description
queen_id	ID of the queen

Request Body

None

Response

Response Body Format

JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No queen with this queen_id exists
Failure	406 Not Acceptable	Returned if an unsupported MIME type is requested. The server will allow for only application/json in the response for a GET request.

Response Examples

Success

```
Status: 200 OK [Content-Type: application/json]

{
    "species": "Western Honey Bee",
    "name": "Eugenie",
    "hive": null,
    "age": 4,
    "id": "5669152601669632",
    "self": "http://my.url/queens/5669152601669632"
}
```

Failure

```
Status: 404 Not Found

{
            "Error":  “No queen with this queen_id exists”
}
```

```
Status: 406 Not Acceptable

{
            "Error": "Unsupported MIME type requested - only application/json supported"
}
```

List all queens (Unprotected)

List all the queen entities currently stored in Datastore. This endpoint is not protected.

```
GET /queens
```

Request

Headers

None

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	Returned if an unsupported MIME type is requested. The server will allow for only application/json in the response for a GET request.

Response Examples

Success

```
Status: 200 OK
{
   "total": 14,
   "queens": [
      {
         "hive": {
            "self": "http://my.url/hives/5075408403824640",
            "id": "5075408403824640"
         },
         "age": 13,
         "name": "Christine",
         "species": "Western Honey Bee",
         "id": "5102723456303104"
      },
      {
         "species": "Western Honey Bee",
         "age": 4,
         "hive": null,
         "name": "Eugenie",
         "id": "5147403464212480"
      },
      {
         "age": 14,
         "hive": null,
         "species": "Western Honey Bee",
         "name": "Alice",
         "id": "5149887163269120"
      },
      {
         "name": "Elizabeth",
         "hive": null,
         "species": "Killer Bee",
         "age": 1,
         "id": "5631671361601536"
      },
      {
         "name": "Fiona",
         "age": 9,
         "species": "Philippine Honey Bee",
         "hive": {
            "self": "http://localhost:8080/hives/4858468934615040",
            "id": "4858468934615040"
         },
         "id": "5643280054222848"
      }
   ],
   "next": "http://my.url/queens?cursor=CjESK2oUZH5wb31ZWVucxiAgIC4iJGDCgwYACAA"
}
```

Failure

```
Status: 406 Not Acceptable

{
	"Error": "Unsupported MIME type requested - only application/json supported"
}
```

Delete a queen (Unprotected)

Allows you to delete a queen. Queens can only be deleted by the authenticated user who created them.

```
DELETE /queens/:queen_id
```

Request

Headers

Header	**Type**	**Contents**
Authorization	Bearer Token	User’s JWT

Path Parameters

Name	**Description**
queen_id	ID of the queen

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No queen with this queen_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

{
        "Error": “No queen with this queen_id exists”
}
```

Update a queen (Unprotected)

Allows you to update any or all base attributes of an existing hive entity. This endpoint is not protected. However, assigning a queen to a hive can only be done by authenticated users.

```
PATCH /queens/:queen_id
```

Request

Headers

Header	Type	Contents
Authorization	Bearer Token	User’s JWT

Path Parameters

Name	Description
queen_id	ID of the queen

Request Body

Required

Request Body Format

JSON (application/json)

Request JSON Attributes

Name	Description	Required?
name	Name given to this queen. Must consist of only valid alphanumeric characters and spaces.	No
species	The species of this bee. Must consist of only valid alphanumeric characters and spaces.	No
age	The approximate age of the queen bee, in months. Type is not verified.	No

Request Body Example

```
{
   "name": "Fiona"
}
```

Response

Response Body Format

JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, if non-alphanumeric characters are used in the name or species fields, the queen is not created and status code 400 is returned.
Failure	404 Not Found	No queen with this queen_id exists
Failure	406 Not Acceptable	Returned if an unsupported MIME type is requested. The server will allow for only application/json in the response for a PATCH request.
Failure	415 Unsupported Media Type	Returned if the client request data is not sent as application/json.

Response Examples

Success

```
Status: 200 OK

Location: http://my.url/queens/5669152601669632

{
    "hive": null,
    "age": 4,
    "name": "Fiona",
    "species": "Western Honey Bee",
    "id": "5669152601669632",
    "self": "http://my.url/queens/5669152601669632"
}
```

Failure

```
Status: 400 Bad Request

{
        "Error":  “Name and species must include only alphanumeric characters”
}
```

```
Status: 404 Not Found

{
        "Error": “No queen with this queen_id exists”
}
```

```
Status: 406 Not Acceptable

{
        "Error": "Unsupported MIME type requested - only application/json supported"
}
```

```
Status: 415 Unsupported Media Type

{
        "Error": "Unsupported MIME type received - server can only accept
        application/json"
}
```

Replace a queen (Unprotected)

Allows you to replace all base attributes of an existing queen entity. This endpoint is not protected. However, assigning a queen to a hive can only be done by authenticated users.

```
PUT /queens/:queen_id
```

Request

Headers

Header	Type	Contents
Authorization	Bearer Token	User’s JWT

Path Parameters

Name	Description
queen_id	ID of the queen

Request Body

Required

Request Body Format

JSON (application/json)

Request JSON Attributes

Name	Description	Required?
name	Name given to this queen. Must consist of only valid alphanumeric characters and spaces.	Yes
species	The species of this bee. Must consist of only valid alphanumeric characters and spaces.	Yes
age	The approximate age of the queen bee, in months. Type is not verified.	Yes

Request Body Example

```
{
   "name": "Elizabeth",
   "species": "Killer Bee",
   "age": 1
}
```

Response

Response Body Format

JSON (application/json)

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	A link to the updated hive page will be placed in the response Location header.
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, if non-alphanumeric characters are used in the name or species fields, the queen is not created and status code 400 is returned.
Failure	404 Not Found	No queen with this queen_id exists
Failure	406 Not Acceptable	Returned if an unsupported MIME type is requested. The server will allow for only application/json in the response for a PATCH request.
Failure	415 Unsupported Media Type	Returned if the client request data is not sent as application/json.
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, if non-alphanumeric characters are used in the name or species fields, the queen is not created and status code 400 is returned.

Response Examples

Success

```
Status: 303 See Other

Location: http://my.url/queens/5669152601669632

{
   "id": 5669152601669632,
   "name": "Elizabeth",
   "species": "Killer Bee",
   "age": 1,
   "hive": null,
   "self": "http://my.url/queens/5669152601669632"
}
```

Failure

```
Status: 400 Bad Request

{
        "Error":  "The request object is missing at least one of the required attributes"
}
```

```
Status: 400 Bad Request

{
        "Error":  “Name and species must include only alphanumeric characters”
}
```

```
Status: 404 Not Found

{
        "Error": “No queen with this queen_id exists”
}
```

```
Status: 406 Not Acceptable

{
        "Error": "Unsupported MIME type requested - only application/json supported"
}
```

```
Status: 415 Unsupported Media Type

{
        "Error": "Unsupported MIME type received - server can only accept
        application/json"
}
```

Assign queen to hive (Protected)

Put the queen with queen_id as the queen assigned to the hive with hive_id. The queen must not be assigned to another hive. Requires the JWT associated with the user who created the hive to be sent in the authorization header for authentication.

```
PUT /hives/:hive_id/queens/:queen_id
```

Request

Headers

Header	Type	Contents
Authorization	Bearer Token	User’s JWT

Path Parameters

Name	Description
hive_id	ID of the hive
queen_id	ID of the queen

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes

Success	204 No Content	Succeeds only if a hive exists with this hive_id, a queen exists with this queen_id, and the queen is not already assigned to another hive.
Failure	401 Unauthorized	If the JWT is missing or invalid, status code 401 is returned.
Failure	403 Forbidden	The queen is already in another hive. It must be removed before it can be assigned to another hive.
Failure	404 Not Found	The given hive_id and/or queen_id do not match any known hive or queen.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized

{
        "Error": “Missing or invalid JWT”
}
```

```
Status: 403 Forbidden

{
        "Error":  “Queen is already assigned”
}
```

```
Status: 404 Not Found


{
        "Error":  “Hive and/or queen not found"
}
```

Remove queen from hive (Protected)

Unqueen the queen with queen_id from the hive with hive_id. Neither are deleted in the process, and the queen can be reassigned upon successful completion.

```
DELETE /hives/:hive_id/queens/:queen_id
```

Request

Headers

Header	Type	Contents
Authorization	Bearer Token	User’s JWT

Path Parameters

Name	Description
hive_id	ID of the hive
queen_id	ID of the queen

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a hive exists with this hive_id, a queen exists with this queen_id, and the queen is assigned to this hive.

Failure	401 Unauthorized	If the JWT is missing or invalid, status code 401 is returned.
Failure	403 Forbidden	The queen is not associated with this hive. It must be removed from the endpoint associated with the correct hive.
Failure	404 Not Found	The given hive_id and/or queen_id do not match any known hive or queen.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized

{
        "Error": “Missing or invalid JWT”
}
```

```
Status: 403 Forbidden

{
        "Error":  “Queen is not associated with this hive”
}
```

```
Status: 404 Not Found


{
        "Error":  “Hive and/or queen not found"
}
```