

Portfolio Project: API Spec

CS 493: Cloud Application Development

Spring 2022

Oregon State University

API URL: <https://portfolio-peifferp.uw.r.appspot.com>

Account Creation URL: <https://portfolio-peifferp.uw.r.appspot.com/login>

Change log.....	1
405 Response Notes	2
Data Model	3
Relationships.....	3
Users	3
Gear.....	3
Rentals	4
User Routes.....	5
Create a User.....	5
Get Users.....	7
Rental Routes.....	10
Create a Rental.....	10
Edit a Rental (Partial)	13
Edit a Rental (Full)	16
Get a Rental	19
Get all Rentals	22
Delete a Rental.....	25
Gear Routes	27
Create Gear	27
Get All Gear	30
Get Piece of Gear	34
Edit Gear (Full)	36
Edit Gear (Partial).....	39
Delete Piece of Gear	42
Rental/Gear Relationship Routes.....	44
Add Piece of Gear to Rental.....	44
Remove Piece of Gear from Rental.....	46
Rental/User Relationship Routes.....	48

Change log

Version	Change	Date
1.0	Initial version.	May 29, 2022

Name: Philip Peiffer

Assignment: Portfolio Project

405 Response Notes

1. 405 Method Not Allowed responses are given for:
 - a. /rentals – DELETE, PUT, PATCH
 - b. /rentals/:rental_id – POST
 - c. /gear – DELETE, PUT, PATCH
 - d. /gear/:gear_id – POST
 - e. /rentals/:rental_id/gear/:gear_id – GET, POST, PATCH

Data Model

The app stores three kinds of entities in Datastore: Customers (users), Gear, and Rentals.

Relationships

Users → Rentals: 1 to Many relationship (a user can have many rentals, but a rental is tied to 1 user)

Rentals → Gear: 1 to Many relationship (a rental can have many pieces of gear, but a certain piece of gear can only be tied to one rental at a time)

Users

Property	Data Type	Req?	Constraints	Notes
First Name	String	Y	Issued by Google account	Obtained through google people API
Last Name	String	Y	Issued by Google account	Obtained through google people API
Date Created	String	Y	Must be in the format MM/DD/YYYY	Auto-created by API when a user is first created.
Rentals	Array	Y	Must be a blank array when customer first created.	Auto-initialized when user is first created. An array showing rentals tied to the user.
Id	String	Y	Issued by Google through the OAuth2.0 process and must match the JWT used to create the user. Must be unique.	This number will match the “sub” category in the JWT for protected requests. This number is also used as the unique ID for user entities.

Gear

Property	Data Type	Req?	Constraints	Notes
item description	String	Y	Must be less than 25 characters and only contain letters of alphabet	A short description of the gear
category	String	Y	Must be one of the categories given in notes	camping, hiking, skiing, snowshoeing, climbing, biking, fishing, water sports
rental	string	Y	Must be null at creation	Auto-initialized to null when gear is created. Only assigned to rental through use of relationship route. Attempts to include in creation request will be rejected.
available	Boolean	Y	Must be true at creation	Auto-initialized to true when gear is first created. When gear is assigned to rental it is changed to false.
id	string	Y	Created by datastore	Auto-created by datastore. Attempts to manually create this value or edit it will be rejected.

Rentals

Property	Data Type	Req?	Constraints	Notes
start	string	Y	Must be in format MM/DD/YYYY	
end	string	Y	Must be in format MM/DD/YYYY	
name	string	N	Must be less than 25 characters and only contain letters of the alphabet	
gear	array	Y	Must be blank on first creation. Otherwise contains array of objects showing gear id and description.	Auto-initialized to an empty array. Attempts to manually create a gear array on rental creation will be rejected.
user	string	Y	Must match JWT provided for rental creation.	Auto-created through the use of a JWT to create a rental. Attempts to hard enter a user value will be rejected.
id	string	Y	Created by datastore.	Auto-created by datastore at time of rental creation. Attempts to hard enter an id will be rejected.

User Routes

Create a User

Allows you to create a new user.

NOTE – this is all handled through the OAuth2.0 process. Once a user grants permission to my app, a new user is created (if they had never logged in before). If a user tries to create a new user through sending a POST request, they will (most likely) be rejected. The app first looks to see if the user id and JWT “sub” property match. If they don’t, the request is rejected. It then looks to see if a user already exists with the given JWT “sub” property. If a user already exists, a new user will not be created but a status of 200 will be returned. If not then a new user is created.

POST /users

Request

Path Parameters

- None

Request Body

- Required? – Yes
- Format? – JSON

Request Authorization

- Required? – Yes
- Type? – JWT
- Header: “Authorization: Bearer Token”

Request JSON Attributes

Name	Description	Required?
First Name	First Name of the user – provided by JWT	Yes
Last Name	Last name of the user – provided by JWT	Yes
Date Created	Date of creation in MM/DD/YYYY format. Auto-created by back end.	Yes
Id	Unique user id – provided by the JWT	Yes

Request Body Example

N/A – this is all handled through OAuth2.0 process. However, internally, the OAuth 2.0 process sends a request with a decoded JWT. The decoded JWT contains the first name, last name, and user ID. Date created is determined from the current date.

Response

Response Body Format

- Success – JSON
- Failure – JSON

Name: Philip Peiffer
Assignment: Portfolio Project

Response Body Notes

- You will not see the following response bodies on creation or failure. You will be redirected back to a page that will display your JWT and User ID for use in the postman collection if your account creation is successful. In the case of your account not being successfully created, you will be redirected to a failure page. The examples given below are just to let you know how the data is stored in datastore.

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	<p>If the request is missing any of the required attributes or is passed more than the required, the user will not be created, and 400 status code will be returned.</p> <p>The API does not validate any of the attribute values and assumes that the values are correct.</p>

Response Examples

Success

Status: 201 Created

```
{
  "First Name": "Philip",
  "Last Name": "Peiffer",
  "Date Created": "05/28/2022"
  "rentals": [],
  "id": "108987079282769582931"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "Request object keys are bad. Check the API doc for required keys."
}
```

Name: Philip Peiffer
Assignment: Portfolio Project

Get Users

Allows you to get a list of all the users.

GET /users

Request

Path Parameters

- None

Request Body

- Required? – No
- Format? – N/A

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- This response is not paginated. If no users exist, a blank array is returned.
- It was not clear on the instructions if the user/entity relationship should be displayed within the user. I chose to display it even though the entity is a protected entity (displaying it represents an authorization violation since anyone can query users and see the list of protected entities assigned to that user). I wouldn't display this info in a real app, but included it for completeness.

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If the "Accept" header is anything other than */* or application/json then the request will be rejected and 406 status returned.

Response Examples

Success

Status: 200 OK

```
[  
  {  
    "First Name": "Philip",
```

Name: Philip Peiffer

Assignment: Portfolio Project

```
"Date Created": "5/28/2022",
"Last Name": "Peiffer",
"rentals": [
  {
    "self": "http://localhost:8080/rentals/6172917368881152",
    "id": "6172917368881152"
  },
  {
    "self": "http://localhost:8080/rentals/5323541280980992",
    "name": "rental #4",
    "id": "5323541280980992"
  },
  {
    "id": "4834670923481088",
    "name": "rental #5",
    "self": "http://localhost:8080/rentals/4834670923481088"
  },
  {
    "id": "4521949657038848",
    "self": "http://localhost:8080/rentals/4521949657038848",
    "name": "rental #6"
  },
  {
    "id": "6089738213654528",
    "name": "rental #7",
    "self": "http://localhost:8080/rentals/6089738213654528"
  }
],
"id": "108987079282769582931"
},
{
  "Date Created": "5/28/2022",
  "Last Name": "Peiffer",
  "First Name": "Philip",
  "rentals": [
    {
      "name": "rental #3",
      "self": "http://localhost:8080/rentals/5609967415459840",
      "id": "5609967415459840"
    }
  ],
  "id": "117242623563545829420"
}
]
```


Name: Philip Peiffer
Assignment: Portfolio Project

Failure

Status: 406 Not Acceptable

```
{  
  "Error": "Server will send back JSON data. Accept header indicates you cannot accept this."  
}
```

Rental Routes

Create a Rental

Allows you to create a new rental.

POST /rentals

Request

Path Parameters

- None

Request Body

- Required? – Yes
- Format? – JSON

Request Authorization

- Required? – Yes
- Type? – JWT
- Header: “Authorization: Bearer Token”

Request JSON Attributes

Name	Description	Required?
start	Date the rental will start	Yes
end	Date the rental will end	Yes
name	A name to remember the rental by	No

Request Body Example

```
{
  "start": "05/27/2022",
  "end": "05/31/2022",
  "name": "Memorial Day Bike Ride"
}
OR (no name)
{
  "start": "05/28/2022",
  "end": "05/31/2022"
}
```

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- Datastore auto generates the ID of the rental, so this value is not passed in as a request parameter.
- User value is determined by the JWT token passed in for authorization, so this value is also not passed in as a request parameter.
- Gear is initialized as a blank array. The array is assigned values through a relationship route.

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the required attributes OR contains more than the maximum attributes, the request is rejected and 400 status is returned. This status is also returned when there is a problem with the format of the JSON body.
Failure	401 Unauthorized	If the request is missing a JWT token or the JWT token is invalid, then the request is rejected and 401 is returned.
Failure	406 Not Acceptable	If the "Accept" header value is not application/json or */* (to indicate anything is accepted), then the request is rejected and 406 is returned.
Failure	415 Unsupported Media Type	If the client tries to send something other than application/json data for the request, then the request is rejected and 415 is returned.

Response Examples

Success

Status: 201 Created

```
{
  "start": "05/27/2022",
  "end": "05/31/2022",
  "name": "Memorial Day Bike Ride",
  "user": "108987079282769582931",
  "gear": [],
  "id": "5647849563881472",
  "self": "http://localhost:8080/rentals/5647849563881472"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "Request object keys are bad. Check the API doc for required keys."
}
OR
{
  "Error": "Problem with JSON format in body."
```

}
<p>Status: 401 Unauthorized</p> <pre>{ "Error": "The request requires a JWT but is either not provided or is invalid." }</pre>
<p>Status: 406 Not Acceptable</p> <pre>{ "Error": "Server will send back JSON data. Accept header indicates you cannot accept this." }</pre>
<p>Status: 415 Unsupported Media Type</p> <pre>{ "Error": "Server can only handle JSON post, put, and patch requests. Please check the request body and content-type header." }</pre>

Name: Philip Peiffer
Assignment: Portfolio Project

Edit a Rental (Partial)

Allows you to partially edit a rental. You CANNOT edit an id or user value. You also CANNOT edit the gear array. This requires calling a relationship route.

PATCH /rentals/:rental_id

Request

Path Parameters

- Rental_id – ID of the rental you wish to edit

Request Body

- Required? – Yes
- Format? – JSON

Request Authorization

- Required? – Yes
- Type? – JWT
- Header: “Authorization: Bearer Token”

Request JSON Attributes

Name	Description	Required?
start	Date the rental will start	No
end	Date the rental will end	No
name	A name to remember the rental by	No

Request Body Example

```
{
  "name": "trying to edit rental3 with PATCH!"
}
```

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	“location” header contains the URI of the updated resource.

Name: Philip Peiffer

Assignment: Portfolio Project

Failure	400 Bad Request	If the request is missing any of the required attributes OR contains more than the maximum attributes, the request is rejected and 400 status is returned. This status is also returned when there is a problem with the format of the JSON body.
Failure	401 Unauthorized	If the request is missing a JWT token or the JWT token is invalid, then the request is rejected and 401 is returned.
Failure	403 Forbidden	If the user ID of the rental does not match the JWT user ID, then the request is rejected and 403 is returned.
Failure	404 Not Found	If the rental does not exist, the request is rejected and 404 is returned.
Failure	406 Not Acceptable	If the "Accept" header value is not application/json or */* (to indicate anything is accepted), then the request is rejected and 406 is returned.
Failure	415 Unsupported Media Type	If the client tries to send something other than application/json data for the request, then the request is rejected and 415 is returned.

Response Examples

Success

Status: 303 See Other

Failure

Status: 400 Bad Request

```
{
  "Error": "Request object keys are bad. Check the API doc for required keys."
}
OR
{
  "Error": "Problem with JSON format in body."
}
```

Status: 401 Unauthorized

```
{
  "Error": "The request requires a JWT but is either not provided or is invalid."
}
```

Status: 403 Forbidden

```
{
  "Error": "JWT provided does not match user of this rental"
}
```

Status: 404 Not Found

```
{  
  "Error": "This rental does not exist."  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Server will send back JSON data. Accept header indicates you cannot accept this."  
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server can only handle JSON post, put, and patch requests. Please check the request body and content-type header."  
}
```

Name: Philip Peiffer
Assignment: Portfolio Project

Edit a Rental (Full)

Allows you to edit a rental. You CANNOT edit an id or user value. You also CANNOT edit the gear array. This requires calling a relationship route.

PUT /rentals/:rental_id

Request

Path Parameters

- Rental_id – ID of the rental you wish to edit

Request Body

- Required? – Yes
- Format? – JSON

Request Authorization

- Required? – Yes
- Type? – JWT
- Header: “Authorization: Bearer Token”

Request JSON Attributes

Name	Description	Required?
start	Date the rental will start. Does not have to be different than existing.	Yes
end	Date the rental will end. Does not have to be different than existing.	Yes
name	A name to remember the rental by. Does not have to be different than existing.	No

Request Body Example

```
{
  "start": "05/30/2022",
  "end": "06/04/2022",
  "name": "trying to edit rental3 with PUT!"
}
```

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	“location” header contains the URI of the updated resource.

Failure	400 Bad Request	If the request is missing any of the required attributes OR contains more than the maximum attributes, the request is rejected and 400 status is returned. This status is also returned when there is a problem with the format of the JSON body.
Failure	401 Unauthorized	If the request is missing a JWT token or the JWT token is invalid, then the request is rejected and 401 is returned.
Failure	403 Forbidden	If the user ID of the rental does not match the JWT user ID, then the request is rejected and 403 is returned.
Failure	404 Not Found	If the rental does not exist, the request is rejected and 404 is returned.
Failure	406 Not Acceptable	If the "Accept" header value is not application/json or */* (to indicate anything is accepted), then the request is rejected and 406 is returned.
Failure	415 Unsupported Media Type	If the client tries to send something other than application/json data for the request, then the request is rejected and 415 is returned.

Response Examples

Success

Status: 303 See Other

Failure

Status: 400 Bad Request

```
{
  "Error": "Request object keys are bad. Check the API doc for required keys."
}
OR
{
  "Error": "Problem with JSON format in body."
}
```

Status: 401 Unauthorized

```
{
  "Error": "The request requires a JWT but is either not provided or is invalid."
}
```

Status: 403 Forbidden

```
{
  "Error": "JWT provided does not match user of this rental"
}
```

Status: 404 Not Found

```
{  
  "Error": "This rental does not exist."  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Server will send back JSON data. Accept header indicates you cannot accept this."  
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server can only handle JSON post, put, and patch requests. Please check the request body and content-type header."  
}
```

Name: Philip Peiffer
Assignment: Portfolio Project

Get a Rental

Allows you to get a single rental.

GET /rentals/:rental_id

Request

Path Parameters

- Rental_id – ID of the rental you wish to see

Request Body

- Required? – No
- Format? – N/A

Request Authorization

- Required? – Yes
- Type? – JWT
- Header: “Authorization: Bearer Token”

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If the request is missing a JWT token or the JWT token is invalid, then the request is rejected and 401 is returned.
Failure	403 Forbidden	If the user ID of the rental does not match the JWT user ID, then the request is rejected and 403 is returned.
Failure	404 Not Found	If the rental does not exist, the request is rejected and 404 is returned.

Name: Philip Peiffer

Assignment: Portfolio Project

Failure	406 Not Acceptable	If the "Accept" header value is not application/json or */* (to indicate anything is accepted), then the request is rejected and 406 is returned.
---------	--------------------	---

Response Examples

Success

Status: 200 OK

```
{
  "end": "05/31/2022",
  "gear": [
    {
      "id": "6248789442560000",
      "item description": "Tent",
      "self": "http://portfolio-peifferp.uw.r.appspot.com/gear/6248789442560000"
    },
    {
      "id": "6251653246222336",
      "item description": "180 Solomon Skis",
      "self": "http://portfolio-peifferp.uw.r.appspot.com/gear/6251653246222336"
    }
  ],
  "user": "108987079282769582931",
  "name": "Memorial Day Bike Ride",
  "start": "05/27/2022",
  "id": "5122889535717376",
  "self": "http://portfolio-peifferp.uw.r.appspot.com/rentals/5122889535717376"
}
```

Failure

Status: 401 Unauthorized

```
{
  "Error": "The request requires a JWT but is either not provided or is invalid."
}
```

Status: 403 Forbidden

```
{
  "Error": "JWT provided does not match user of this rental"
}
```

Status: 404 Not Found

```
{  
  "Error": "This rental does not exist."  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Server will send back JSON data. Accept header indicates you cannot accept this."  
}
```

Name: Philip Peiffer
Assignment: Portfolio Project

Get all Rentals

Allows you to get all rentals corresponding to the user whose ID matches the JWT passed in the request.

GET /rentals

Request

Path Parameters

- None

Request Body

- Required? – No
- Format? – N/A

Request Authorization

- Required? – Yes
- Type? – JWT
- Header: “Authorization: Bearer Token”

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- The response is paginated to 5 results per page.

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If the request is missing a JWT token or the JWT token is invalid, then the request is rejected and 401 is returned.
Failure	406 Not Acceptable	If the “Accept” header value is not application/json or */* (to indicate anything is accepted), then the request is rejected and 406 is returned.

Response Examples

Success

Status: 200 OK

Name: Philip Peiffer

Assignment: Portfolio Project

```
{
  "next": "http://portfolio-
peifferp.uw.r.appspot.com/rentals?token=CjQSLmoWenV3fnBvcnRmb2xpbylwZWlmZmVycHIUCxIHcm
VudGFscxiAgIDY%2BIaPCgwYACAA",
  "total": 6,
  "rentals": [
    {
      "end": "06/04/2022",
      "start": "05/30/2022",
      "user": "108987079282769582931",
      "gear": [
        {
          "id": "6248789442560000",
          "item description": "Tent",
          "self": "http://portfolio-
peifferp.uw.r.appspot.com/gear/6248789442560000"
        },
        {
          "item description": "180 Solomon Skis",
          "id": "6251653246222336",
          "self": "http://portfolio-
peifferp.uw.r.appspot.com/gear/6251653246222336"
        }
      ],
      "name": "trying to edit rental1!",
      "id": "5122889535717376",
      "self": "http://portfolio-
peifferp.uw.r.appspot.com/rentals/5122889535717376"
    },
    {
      "name": "rental #7",
      "end": "05/32/2022",
      "start": "05/28/2022",
      "gear": [],
      "user": "108987079282769582931",
      "id": "5129025567588352",
      "self": "http://portfolio-
peifferp.uw.r.appspot.com/rentals/5129025567588352"
    },
    {
      "gear": [],
      "user": "108987079282769582931",
      "end": "05/32/2022",
      "start": "05/28/2022",
      "name": "rental #6",
      "id": "5202293146255360",
      "self": "http://portfolio-
```

```
peifferp.uw.r.appspot.com/rentals/5202293146255360"
    },
    {
      "start": "05/28/2022",
      "user": "108987079282769582931",
      "name": "rental #5",
      "gear": [],
      "end": "05/32/2022",
      "id": "5691975521009664",
      "self": "http://portfolio-
peifferp.uw.r.appspot.com/rentals/5691975521009664"
    },
    {
      "user": "108987079282769582931",
      "name": "rental #4",
      "gear": [],
      "start": "05/28/2022",
      "end": "05/32/2022",
      "id": "5695708787113984",
      "self": "http://portfolio-
peifferp.uw.r.appspot.com/rentals/5695708787113984"
    }
  ]
}
```

Failure

Status: 401 Unauthorized

```
{
  "Error": "The request requires a JWT but is either not provided or is invalid."
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server will send back JSON data. Accept header indicates you cannot accept this."
}
```


Name: Philip Peiffer
Assignment: Portfolio Project

Delete a Rental

Allows you to delete a single rental. NOTE – the rental is removed from any gear in the gear array as well.

DELETE /rentals/:rental_id

Request

Path Parameters

- Rental_id – ID of the rental you wish to see

Request Body

- Required? – No
- Format? – N/A

Request Authorization

- Required? – Yes
- Type? – JWT
- Header: “Authorization: Bearer Token”

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – None
- Failure – JSON

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	If the request is missing a JWT token or the JWT token is invalid, then the request is rejected and 401 is returned.
Failure	403 Forbidden	If the user ID of the rental does not match the JWT user ID, then the request is rejected and 403 is returned.
Failure	404 Not Found	If the rental does not exist, the request is rejected and 404 is returned.

Name: Philip Peiffer
Assignment: Portfolio Project

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{  
  "Error": "The request requires a JWT but is either not provided or is invalid."  
}
```

Status: 403 Forbidden

```
{  
  "Error": "JWT provided does not match user of this rental"  
}
```

Status: 404 Not Found

```
{  
  "Error": "This rental does not exist."  
}
```

Gear Routes

Create Gear

Allows you to create a piece of gear. You CANNOT add an id, available, or rental value to the request. These are automatically initialized to their correct values.

POST /gear

Request

Path Parameters

- None

Request Body

- Required? – Yes
- Format? – JSON

Request Authorization

- Required? – No
- Type? – N/A
- Header: N/A

Request JSON Attributes

Name	Description	Required?
item description	A description of the piece of gear	Yes
category	The category that you would find it in a store. See constraints towards beginning for allowed categories.	Yes

Request Body Example

```
{
  "item description": "Tent",
  "category": "camping"
}
```

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- Rental, available, and id are automatically initialized to their correct values of null, true, and whatever datastore assigns, respectively.

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	

Failure	400 Bad Request	If the request is missing any of the required attributes OR contains more than the maximum attributes, the request is rejected and 400 status is returned. This status is also returned when there is a problem with the format of the JSON body.
Failure	406 Not Acceptable	If the "Accept" header value is not application/json or */* (to indicate anything is accepted), then the request is rejected and 406 is returned.
Failure	415 Unsupported Media Type	If the client tries to send something other than application/json data for the request, then the request is rejected and 415 is returned.

Response Examples

Success

Status: 201 Created

```
{
  "item description": "Tent",
  "category": "camping",
  "rental": null,
  "available": true,
  "id": "6210799517302784",
  "self": "http://localhost:8080/gear/6210799517302784"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "Request object keys are bad. Check the API doc for required keys."
}
OR
{
  "Error": "Problem with JSON format in body."
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server will send back JSON data. Accept header indicates you cannot accept this."
}
```

Name: Philip Peiffer

Assignment: Portfolio Project

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server can only handle JSON post, put, and patch requests. Please check  
the request body and content-type header."  
}
```

Name: Philip Peiffer
Assignment: Portfolio Project

Get All Gear

Allows you to get all of the gear currently in the database.

GET /gear

Request

Path Parameters

- token – optional, passed in when using cursor to get to next page of results. Defaults to null.

Request Body

- Required? – No
- Format? – N/A

Request Authorization

- Required? – No
- Type? – N/A
- Header: N/A

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- Self is created on the fly by the backend.
- Results are paginated to 5 results per page.
- If “next” is null, then you’ve reached the end of the results.
- “Total” contains the total number of items currently in the database.

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	If the piece of gear does not exist in the datastore, 404 status is returned.
Failure	406 Not Acceptable	If the “Accept” header value is not application/json or */* (to indicate anything is accepted), then the request is rejected and 406 is returned.

Name: Philip Peiffer

Assignment: Portfolio Project

Response Examples

Success

Status: 200 OK

```
{
  "next": "http://localhost:8080/gear?token=CjESK2oWenV3fnBvcnRmb2xpbylwZWlmZmVycHIR
CxIEZ2VhchiAgICYh5aECwwYACAA",
  "total": 6,
  "gear": [
    {
      "rental": null,
      "category": "skiing",
      "item description": "180 Solomon Skis",
      "available": true,
      "id": "4675146040213504",
      "self": "http://localhost:8080/gear/4675146040213504"
    },
    {
      "available": true,
      "item description": "180 Solomon Skis",
      "rental": null,
      "category": "skiing",
      "id": "4803424633749504",
      "self": "http://localhost:8080/gear/4803424633749504"
    },
    {
      "category": "biking",
      "available": true,
      "item description": "Gear #5",
      "rental": null,
      "id": "5526788260233216",
      "self": "http://localhost:8080/gear/5526788260233216"
    },
    {
      "rental": null,
      "item description": "Gear #3",
      "available": true,
      "category": "water sports",
      "id": "5801045947056128",
      "self": "http://localhost:8080/gear/5801045947056128"
    },
    {
      "available": true,
      "rental": null,
      "item description": "Gear #1 PUT",
      "category": "biking",
      "id": "6210799517302784",
      "self": "http://localhost:8080/gear/6210799517302784"
    }
  ]
}
```


Name: Philip Peiffer
Assignment: Portfolio Project

Failure

Status: 406 Not Acceptable

```
{  
  "Error": "Server will send back JSON data. Accept header indicates you cannot accept this."  
}
```

Name: Philip Peiffer
Assignment: Portfolio Project

Get Piece of Gear

Allows you to get a piece of gear.

GET /gear/:gear_id

Request

Path Parameters

- gear_id – id of the gear you wish to view

Request Body

- Required? – No
- Format? – N/A

Request Authorization

- Required? – No
- Type? – N/A
- Header: N/A

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- Self is created on the fly by the backend.

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	If the piece of gear does not exist in the datastore, 404 status is returned.
Failure	406 Not Acceptable	If the “Accept” header value is not application/json or */* (to indicate anything is accepted), then the request is rejected and 406 is returned.

Name: Philip Peiffer
Assignment: Portfolio Project

Response Examples

Success

Status: 200 OK

```
{
  "item description": "Tent",
  "category": "camping",
  "rental": null,
  "available": true,
  "id": "6210799517302784",
  "self": "http://localhost:8080/gear/6210799517302784"
}
```

Failure

Status: 404 Not Found

```
{
  "Error": "This gear does not exist."
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server will send back JSON data. Accept header indicates you cannot accept this."
}
```

Name: Philip Peiffer
Assignment: Portfolio Project

Edit Gear (Full)

Allows you to edit a piece of gear. You CANNOT edit an id, available, or rental value. To edit available and rental requires calling a relationship route. Note that if item description is changed, then the rental associated with the gear is also changed to reflect the new description.

PUT /gear/:gear_id

Request

Path Parameters

- gear_id – ID of the gear you wish to edit

Request Body

- Required? – Yes
- Format? – JSON

Request Authorization

- Required? – No
- Type? – N/A
- Header: N/A

Request JSON Attributes

Name	Description	Required?
item description	A description of the piece of gear	Yes
category	The category that you would find it in a store. See constraints towards beginning for allowed categories.	Yes

Request Body Example

```
{
  "item description": "Gear #1 PUT",
  "category": "biking"
}
```

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	"location" header contains the URI of the updated resource.

Name: Philip Peiffer

Assignment: Portfolio Project

Failure	400 Bad Request	If the request is missing any of the required attributes OR contains more than the maximum attributes, the request is rejected and 400 status is returned. This status is also returned when there is a problem with the format of the JSON body.
Failure	404 Not Found	If the rental does not exist, the request is rejected and 404 is returned.
Failure	406 Not Acceptable	If the "Accept" header value is not application/json or */* (to indicate anything is accepted), then the request is rejected and 406 is returned.
Failure	415 Unsupported Media Type	If the client tries to send something other than application/json data for the request, then the request is rejected and 415 is returned.

Response Examples

Success

Status: 303 See Other

Failure

Status: 400 Bad Request

```
{
  "Error": "Request object keys are bad. Check the API doc for required keys."
}
OR
{
  "Error": "Problem with JSON format in body."
}
```

Status: 404 Not Found

```
{
  "Error": "This gear does not exist."
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server will send back JSON data. Accept header indicates you cannot accept this."
}
```

Name: Philip Peiffer

Assignment: Portfolio Project

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server can only handle JSON post, put, and patch requests. Please check  
the request body and content-type header."  
}
```

Name: Philip Peiffer
Assignment: Portfolio Project

Edit Gear (Partial)

Allows you to edit a piece of gear. You CANNOT edit an id, available, or rental value. To edit available and rental requires calling a relationship route. Note that if item description is changed, then the rental associated with the gear is also changed to reflect the new description.

PATCH /gear/:gear_id

Request

Path Parameters

- gear_id – ID of the gear you wish to edit

Request Body

- Required? – Yes
- Format? – JSON

Request Authorization

- Required? – No
- Type? – N/A
- Header: N/A

Request JSON Attributes

Name	Description	Required?
item description	A description of the piece of gear	No
category	The category that you would find it in a store. See constraints towards beginning for allowed categories.	No

Request Body Example

```
{  
  "item description": "Gear #1 PATCH"  
}
```

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	"location" header contains the URI of the updated resource.

Failure	400 Bad Request	If the request is missing any of the required attributes OR contains more than the maximum attributes, the request is rejected and 400 status is returned. This status is also returned when there is a problem with the format of the JSON body.
Failure	404 Not Found	If the rental does not exist, the request is rejected and 404 is returned.
Failure	406 Not Acceptable	If the "Accept" header value is not application/json or */* (to indicate anything is accepted), then the request is rejected and 406 is returned.
Failure	415 Unsupported Media Type	If the client tries to send something other than application/json data for the request, then the request is rejected and 415 is returned.

Response Examples

Success

Status: 303 See Other

Failure

Status: 400 Bad Request

```
{
  "Error": "Request object keys are bad. Check the API doc for required keys."
}
OR
{
  "Error": "Problem with JSON format in body."
}
```

Status: 404 Not Found

```
{
  "Error": "This gear does not exist."
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server will send back JSON data. Accept header indicates you cannot accept this."
}
```


Name: Philip Peiffer

Assignment: Portfolio Project

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server can only handle JSON post, put, and patch requests. Please check  
the request body and content-type header."  
}
```

Delete Piece of Gear

Allows you to delete a piece of gear. Note that this will also remove the gear from any rental that it is currently tied to by removing it from the gear array on the rental.

DELETE /gear/:gear_id

Request

Path Parameters

- gear_id – id of the gear you wish to delete

Request Body

- Required? – No
- Format? – N/A

Request Authorization

- Required? – No, I realize this is a quirk because you could technically alter a protected resource (rentals) without authorization. In a real app, I would call the protected relationship route, passing in any JWT provided to break the relationship and then delete the resource.
- Type? – N/A
- Header: N/A

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – None
- Failure – JSON

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	If the piece of gear does not exist in the datastore, 404 status is returned.

Response Examples

Success

Status: 204 No Content

Name: Philip Peiffer

Assignment: Portfolio Project

Failure

Status: 404 Not Found

```
{  
  "Error": "This gear does not exist."  
}
```

Rental/Gear Relationship Routes

Add Piece of Gear to Rental

Allows you to add a piece of gear to a rental. Changes the following for gear: available=false, rental=rental_id. Changes the following for rental: gear.add({"id": gear_id, "item description": gear.item_description})

PUT /rental/:rental_id/gear/:gear_id

Request

Path Parameters

- gear_id – id of the gear you wish to tie to the rental
- rental_id – id of the rental you'd like to tie the gear to

Request Body

- Required? – No
- Format? – N/A

Request Authorization

- Required? – Yes
- Type? – JWT
- Header: "Authorization: Bearer <Token>"

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	This status is returned if the gear is currently unavailable (tied to another rental).
Failure	401 Unauthorized	If the JWT is missing or is invalid, 403 status is returned.
Failure	403 Forbidden	If the owner of the JWT does not own the rental, then the request is rejected and 403 status is returned.

Failure	404 Not Found	If the piece of gear or rental do not exist in the datastore, 404 status is returned.
---------	---------------	---

Response Examples

Success

Status: 204 No Content

Failure

Status: 400 Bad Request { "Error": "This gear is already rented." }
Status: 401 Unauthorized { "Error": "The request requires a JWT but is either not provided or is invalid." }
Status: 403 Forbidden { "Error": "JWT provided does not match user of this rental" }
Status: 404 Not Found { "Error": "This gear does not exist." } OR { "Error": "This rental does not exist." }

Name: Philip Peiffer
Assignment: Portfolio Project

Remove Piece of Gear from Rental

Allows you to remove a piece of gear from a rental. Changes the following for gear: available=true, rental=null. Changes the following for rental: gear.remove({"id": gear_id, "item description": gear.item_description})

DELETE /rental/:rental_id/gear/:gear_id

Request

Path Parameters

- gear_id – id of the gear you wish to remove from the rental
- rental_id – id of the rental you'd like to remove the gear from

Request Body

- Required? – No
- Format? – N/A

Request Authorization

- Required? – Yes
- Type? – JWT
- Header: "Authorization: Bearer <Token>"

Request JSON Attributes

Name	Description	Required?
N/A	N/A	N/A

Request Body Example

N/A

Response

Response Body Format

- Success – JSON
- Failure – JSON

Response Body Notes

- None

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	This status is returned if the gear and rental exist but are not related.
Failure	401 Unauthorized	If the JWT is missing or is invalid, 403 status is returned.
Failure	403 Forbidden	If the owner of the JWT does not own the rental, then the request is rejected and 403 status is returned.
Failure	404 Not Found	If the piece of gear or rental do not exist in the datastore, 404 status is returned.

Name: Philip Peiffer
Assignment: Portfolio Project

Response Examples

Success

Status: 204 No Content

Failure

Status: 400 Bad Request

```
{  
  "Error": "The gear is not on this rental"  
}
```

Status: 401 Unauthorized

```
{  
  "Error": "The request requires a JWT but is either not provided or is invalid."  
}
```

Status: 403 Forbidden

```
{  
  "Error": "JWT provided does not match user of this rental"  
}
```

Status: 404 Not Found

```
{  
  "Error": "This gear does not exist."  
}  
OR  
{  
  "Error": "This rental does not exist."  
}
```

Rental/User Relationship Routes

All of these routes are handled automatically through the use of a JWT. Therefore, no extra routes are given for:

- Assigning a rental to a user (handled with route POST /rentals)
- Deleting a rental from a user (handled with DELETE /rentals/:rental_id)

Having additional routes didn't make sense to me because you're essentially automatically defining the route /users/:user_id through the use of a JWT, so every request to a protected resource would look like:

/users/:user_id/protected_resource/:protected_resource_id