

The URL where your application is deployed on GCP.

- <https://final-suhs.appspot.com/>

The URL for account creation/login (can be the same as above).

- <https://final-suhs.appspot.com/>

Final Project: API Spec

CS 493: Cloud Application Development

Spring 2021

Oregon State University

Last Update: June 6th, 2021

Data Model	Error! Bookmark not defined.
List All Users	4
Create a Group	5
List all Groups	7
Delete a Group	9
Edit a Subset of a Group	10
Edit all Attributes of a Group	12
Create a Post	14
List all Posts	17
Delete a Post	19
Edit a Subset of a Post	20
Edit all Attributes of a Post	23
Assign a Group to a Post	26
Remove a Group from a Post	28
Create a Comment	30
List all Comments	32
Delete a Comment	33
Edit a Subset of a Comment	34
Edit all Attributes of a Comment	37

Data Model

Users

Property Name	Type	Required?	Notes
id	String	No	The unique id of user. This is generated automatically when a user logs in or creates an account. This is equivalent to the “sub” value from the JWT.
email	String	No	The email of the logged in user. This is automatically fetched from the user after login/signup.

Groups *(unprotected)*

Property Name	Type	Required?	Notes
id	String	No	The unique id of group. This is automatically generated by Datastore.
name	String	Yes	The name of the group. Must be a string.
description	String	Yes	The description of group. Must be a string.
creation_date	String	No	Creation Date is automatically generated.
posts	Array of posts	No	Array of posts of a group.

Posts *(protected)*

Property Name	Type	Required?	Notes
id	String	No	The unique id of post. This is automatically generated by Datastore.
author	String	No	The author of the post. This is automatically generated based on the JWT.
group	String	No	The group the post belongs to. This is optional and may be assigned/unassigned later on. Default: None.
title	String	Yes	Title of the post. Must be a string.
body	String	Yes	Body of the post. Must be a String.
creation_date	String	No	Creation Date is automatically generated.

last_modified	String	No	Last Modified Date is automatically generated during creation. Updated when the contents of the post are updated.
comments	Array of comments	No	Array of comments of a post.

Comments *(protected)*

Property Name	Type	Required?	Notes
id	String	No	The unique id of comment. This is automatically generated by Datastore.
author	String	No	The author of the comment. This is automatically generated based on the JWT.
body	String	Yes	Body of the comment. Must be a String.
creation_date	String	No	Creation Date is automatically generated.
last_modified	String	No	Last Modified Date is automatically generated during creation. Updated when the contents of the comment are updated.
post_id	String	No	The post_id of the post that the comment belongs to.

Non-User Entities Relationship

Groups and Posts

- Groups and Posts have an optional relationship.
- A post can be created with/without a group.
- The group of a post can be assigned/unassigned.
- When a group is deleted, the group of the post is updated to “None”.

Posts and Comments

- Posts and Comments have a mandatory relationship.
- A comment has to be created to a specific post.
- When a post is deleted, all comments for that post is also deleted.

User Entity

- The unique identifier for the users is equivalent to the “sub” value from the JWT.
- The supplied JWT of a request is matched to this unique identifier.
- A user can directly create posts and comments.
- Only the owner of a post/comment can read, create, update, and delete a post/comment.

List all Users

List all the users.

GET /users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK
<pre>{ "email": "abc@gmail.com", "id": "google-oauth2 100808511338155876176" }, { "email": "def@gmail.com", "id": "google-oauth2 103346317843634340680" }</pre>

Create a Group *Unprotected*

Allows you to create a new group.

POST /groups

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the group.	Yes
description	String	The description of the group.	Yes

Request Body Example

Accepted <pre>{ "name": "Movies", "description": "Place to discuss movies", }</pre>
Unaccepted (missing at least one attribute) <pre>{ "name": "Movies" }</pre>
Unaccepted (not JSON Format) <pre>name: Movies description: Place to discuss movies</pre>

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Only “name” and “description” will be added to the new entity. Any other attribute will be ignored.
Failure	400 Bad Request	If the request is missing any of the 2 required attributes, the group must not be created, and 400 status code must be returned.
Failure	406 Not Acceptable	Response Format is in JSON. If client cannot accept JSON, then group is not created, and 406 status code is returned.
Failure	415 Unsupported Media Type	Request Format must be in JSON. If client’s request format is not in JSON, group is not created, and 415 status code is returned.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The `self` attribute will contain the live link to the REST resource corresponding to this group. In other words, this is the URL to get this newly created group.
- Any other attributes in the request will be ignored and will not be added to the entity.

Success

```
Status: 201 Created
{
  "id": "abc123",
  "name": "Movies",
  "description": "Place to discuss movies",
  "creation_date": "06/06/2021 22:59:49",
  "posts": [],
  "self": "https://final-suhs.appspot.com/groups/abc123"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

```
{
  "Error": "Name is not a string"
}
```

```
{
  "Error": "Description is not a string"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Server can only return application/json"
}
```

```
Status: 415 Unsupported Media Type
{
  "Error": "Request is an unsupported media type. Request must be JSON"
}
```

List all Groups *Protected*

List all the groups.

GET /groups

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

```
Status: 200 OK
{
  "total_groups": 6,
  "groups": [
    {
      "posts": [],
      "name": "Music",
      "description": "La la la",
      "create_date": "06/06/2021 23:34:58",
      "id": 4673484156305408,
      "self": "https://final-suhs.appspot.com/groups/4673484156305408"
    },
    {
      "posts": [],
      "description": "Gossip",
      "name": "Celebrity",
      "create_date": "06/06/2021 23:35:01",
      "id": 4675146040213504,
      "self": "https://final-suhs.appspot.com/groups/4675146040213504"
    },
    {
      "posts": [],
      "description": "Technology is cool",
      "name": "Tech",
      "create_date": "06/06/2021 23:35:04",
      "id": 4733584338321408,
      "self": "https://final-suhs.appspot.com/groups/4733584338321408"
    },
    {
      "name": "Movies",
      "posts": [],

```

```
    "description": "Place to talk about movies",
    "create_date": "06/06/2021 22:59:49",
    "id": 6499799276716032,
    "self": "https://final-suhs.appspot.com/groups/6499799276716032"
  },
  {
    "name": "Food",
    "posts": [],
    "create_date": "06/06/2021 23:34:59",
    "description": "Yummy",
    "id": 6504733388832768,
    "self": "https://final-suhs.appspot.com/groups/6504733388832768"
  }
],
"next": "https://final-suhs.appspot.com/groups?limit=5&offset=5"
}
```


Delete a Group *Unprotected*

Allows you to delete a group.

DELETE /groups/:group_id

Request

Path Parameters

Name	Type	Description
group_id	String	ID of the group

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No group with this group_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found
<pre>{ "Error": "No group with this group_id exists" }</pre>

Edit a Subset of a Group *Unprotected*

Allows you to edit a subset of a group's attributes

PATCH /groups/:group_id

Request

Path Parameters

Name	Type	Description
group_id	String	ID of the group

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the group.	No
description	String	The description of the group.	No
At least one of the two attributes are required.			

Request Body Example

```
{
  "name": "New Movies"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing all of the 2 required attributes but body is in JSON format with other attributes included, the group not be created, and 400 status code must be returned.
Failure	404 Not Found	No group with this group_id exists
Failure	406 Not Acceptable	If the request is an empty body, the group must not be created, and 406 status code must be returned. Response Format is in JSON. If client cannot accept JSON, then group is not created, and 406 status code is returned.
Failure	415 Unsupported Media Type	Request Format must be in JSON. If client's request format is not in JSON, group is not created, and 415 status code is returned.

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this group. In other words, this is the URL to get this newly created group.
- Any other attributes in the request will be ignored and will not be added to the entity.

Response Examples

Success

```
Status: 200 OK
{
  "id": "abc123",
  "name": "New Movies",
  "description": "Place to discuss movies",
  "creation_date": "06/06/2021 22:59:49",
  "posts": [],
  "self": "https://final-suhs.appspot.com/groups/abc123"
}
```

Failure

```
Status: 400 Bad Request

{
  "Error": "No attribute was given"
}

{
  "Error": "Name is not a string"
}

{
  "Error": "Description is not a string"
}
```

```
Status: 404 Not Found

{
  "Error": "No group with this group_id exists"
}
```

```
Status: 406 Not Acceptable

{
  "Error": "Server can only return application/json"
}
```

```
Status: 415 Unsupported Media Type

{
  "Error": "Request is an unsupported media type. Request must be JSON"
}
```

Edit all Attributes of a Group *Unprotected*

Allows you to edit all attributes of a group

PUT /groups/:group_id

Request

Path Parameters

Name	Type	Description
group_id	String	ID of the group

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the group.	Yes
description	String	The description of the group.	Yes
All two attributes are required.			

Request Body Example

```
{
  "name": "New Movies",
  "type": "New place to discuss movies"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	303 Status code is returned upon success. The URL of the updated group will be included in the Location header.
Failure	400 Bad Request	If the request is missing all of the two required attributes, the group must not be created, and 400 status code must be returned.
Failure	404 Not Found	No group with this group_id exists
Failure	406 Not Acceptable	Response Format is in JSON. If client cannot accept JSON, then group is not created, and 406 status code is returned.
Failure	415 Unsupported Media Type	Request Format must be in JSON. If client's request format is not in JSON, group is not created, and 415 status code is returned.

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this group. In other words, this is the URL to get this newly created group.

- Any other attributes in the request will be ignored and will not be added to the entity.

Response Examples

Success

Status: 303 See Other

```
{
  "id": "abc123",
  "name": "New Movies",
  "description": "New place to discuss movies",
  "creation_date": "06/06/2021 22:59:49",
  "posts": [],
  "self": "https://final-suhs.appspot.com/groups/abc123"
}
```

Failure

Status: 400 Bad Request

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

```
{
  "Error": "Name is not a string"
}
```

```
{
  "Error": "Description is not a string"
}
```

Status: 404 Not Found

```
{
  "Error": "No group with this group_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server can only return application/json"
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "Request is an unsupported media type. Request must be JSON"
}
```

Create a Post *Protected*

Allows you to create a new post.

POST /posts

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
group	String	The group the post belongs to.	No
title	String	The title of the post.	Yes
body	String	The body of the post.	Yes

Request Body Example

Accepted <pre>{ "title": "I love movies", "body": "I want to watch more movies", }</pre>
Unaccepted (missing at least one attribute) <pre>{ "title": "I love movies" }</pre>
Unaccepted (not JSON Format) title: I love movies body: I want to watch more movies

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Group is defaulted to "None". Only "group", "title", and "body" will be added to the new entity. Any other attribute will be ignored.
Failure	400 Bad Request	If the request is missing any of the 2 required attributes, the post must not be created, and 400 status code must be returned.
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.
Failure	406 Not Acceptable	Response Format is in JSON. If client cannot accept JSON, then post is not created, and 406 status code is returned.

Failure	415 Unsupported Media Type	Request Format must be in JSON. If client's request format is not in JSON, post is not created, and 415 status code is returned.
---------	----------------------------	--

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The `self` attribute will contain the live link to the REST resource corresponding to this group. In other words, this is the URL to get this newly created group.
- Any other attributes in the request will be ignored and will not be added to the entity.

Success

```
Status: 201 Created
{
  "id": "abc123",
  "group": "NULL",
  "author": "google-oauth2|103346317843634340680",
  "title": "I love movies",
  "body": "I want to watch more movies",
  "creation_date": "06/06/2021 22:59:49",
  "last_modified": "06/06/2021 22:59:49",
  "comments": [],
  "self": "https://final-suhs.appspot.com/posts/abc123"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}

{
  "Error": "title is not a string"
}

{
  "Error": "body is not a string"
}

Status: 401 Unauthorized
{
  "Error": "Invalid JWT"
}
{
  "Error": "Missing JWT"
}

Status: 406 Not Acceptable
{
  "Error": "Server can only return application/json"
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Request is an unsupported media type. Request must be JSON"  
}
```


List all Posts *Protected*

List all the posts.

GET /posts

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.

Response Examples

Success

<pre>Status: 200 OK { "total_posts": 2, "posts": [{ "title": "Anyone want to watch a movie?", "group": null, "creation_date": "06/06/2021 23:36:22", "author": "google-oauth2 103346317843634340680", "comments": [], "body": "What should we watch?", "last_modified": "06/06/2021 23:36:22", "id": 5799384063148032, "self": "https://final-suhs.appspot.com/posts/5799384063148032" }, { "author": "google-oauth2 103346317843634340680", "comments": [], "creation_date": "06/06/2021 23:08:21", "title": "Anyone want to watch a movie?", "group": null, "last_modified": "06/06/2021 23:08:21", "body": "What should we watch?", "id": 4714247422476288, "self": "https://final-suhs.appspot.com/posts/4714247422476288" }] }</pre>

Failure

Status: 401 Unauthorized

```
{
```

```
  "Error": "Invalid JWT"
```

```
}
```

```
{
```

```
  "Error": "Missing JWT"
```

```
}
```

Delete a Post *Protected*

Allows you to delete a post.

DELETE /posts/:post_id

Request

Path Parameters

Name	Type	Description
post_id	String	ID of the post

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.
Failure	403 Forbidden	If user is not the author of the post, 403 code is returned.
Failure	404 Not Found	No post with this post_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized { "Error": "Invalid JWT" }
{ "Error": "Missing JWT" }
Status: 403 Forbidden { "Error": "Post is owned by someone else" }
Status: 404 Not Found { "Error": "No post with this post_id exists" }

Edit a Subset of a Post *Protected*

Allows you to edit a subset of a post's attributes

PATCH /posts/:post_id

Request

Path Parameters

Name	Type	Description
post_id	String	ID of the post

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
group	String	Group the post belongs to	No
title	String	The title of the post.	No
body	String	The body of the post.	No
At least one of the three attributes are required.			

Request Body Example

```
{
  "title": "Yes, I love movies"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing all of the 3 required attributes but body is in JSON format with other attributes included, the post not be created, and 400 status code must be returned.
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.
Failure	403 Forbidden	If user is not the author of the post, 403 code is returned.
Failure	404 Not Found	No post with this post_id exists
Failure	406 Not Acceptable	If the request is an empty body, the post must not be created, and 406 status code must be returned. Response Format is in JSON. If client cannot accept JSON, then post is not created, and 406 status code is returned.
Failure	415 Unsupported Media Type	Request Format must be in JSON. If client's request format is not in JSON, post is not created, and 415 status code is returned.

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this post. In other words, this is the URL to get this newly created post.
- Any other attributes in the request will be ignored and will not be added to the entity.

Response Examples

Success

```
Status: 200 OK
{
  "id": "abc123",
  "group": "NULL",
  "author": "google-oauth2|103346317843634340680",
  "title": "Yes, I love movies",
  "body": "I want to watch more movies",
  "creation_date": "06/06/2021 22:59:49",
  "last_modified": "06/06/2021 23:02:21",
  "comments": [],
  "self": "https://final-suhs.appspot.com/posts/abc123"
}
```

Failure

```
Status: 400 Bad Request

{
  "Error": "No attribute was given"
}

{
  "Error": "Group is not a string"
}

{
  "Error": "Title is not a string"
}

{
  "Error": "Body is not a string"
}
```

```
Status: 401 Unauthorized
{
  "Error": "Invalid JWT"
}
{
  "Error": "Missing JWT"
}
```

Status: 403 Forbidden { "Error": "Post is owned by someone else" }
Status: 404 Not Found { "Error": "No post with this post_id exists" }
Status: 406 Not Acceptable { "Error": "Server can only return application/json" }
Status: 415 Unsupported Media Type { "Error": "Request is an unsupported media type. Request must be JSON" }

Edit all Attributes of a Post *Protected*

Allows you to edit all attributes of a post

PUT /posts/:post_id

Request

Path Parameters

Name	Type	Description
post_id	String	ID of the post

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
group	String	Group the post belongs to	No
title	String	The title of the post.	Yes
body	String	The body of the post.	Yes
"title" and "body" attributes are required. "group" is not required due to nature of relationship.			

Request Body Example

```
{
  "title": "Yes, I love movies",
  "body": "Yes, I want to watch more movies"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	303 Status code is returned upon success. The URL of the updated post will be included in the Location header.
Failure	400 Bad Request	If the request is missing all of the two required attributes, the post must not be created, and 400 status code must be returned.
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.
Failure	403 Forbidden	If user is not the author of the post, 403 code is returned.
Failure	404 Not Found	No post with this post_id exists
Failure	406 Not Acceptable	Response Format is in JSON. If client cannot accept JSON, then post is not created, and 406 status code is returned.
Failure	415 Unsupported Media Type	Request Format must be in JSON. If client's request format is not in JSON, post is not created, and 415 status code is returned.

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this post. In other words, this is the URL to get this newly created post.
- Any other attributes in the request will be ignored and will not be added to the entity.

Response Examples

Success

Status: 303 See Other

```
{
  "id": "abc123",
  "group": "NULL",
  "author": "google-oauth2|103346317843634340680",
  "title": "Yes, I love movies",
  "body": "Yes, I want to watch more movies",
  "creation_date": "06/06/2021 22:59:49",
  "last_modified": "06/06/2021 23:02:21",
  "comments": [],
  "self": "https://final-suhs.appspot.com/posts/abc123"
}
```

Failure

Status: 400 Bad Request

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

```
{
  "Error": "Group is not a string"
}
```

```
{
  "Error": "Title is not a string"
}
```

```
{
  "Error": "Body is not a string"
}
```

Status: 401 Unauthorized

```
{
  "Error": "Invalid JWT"
}
```

```
{
  "Error": "Missing JWT"
}
```

Status: 403 Forbidden

```
{
```


<pre>"Error": "Post is owned by someone else"</pre>
<pre>}</pre>
<pre>Status: 404 Not Found</pre>
<pre>{</pre>
<pre>"Error": "No post with this post_id exists"</pre>
<pre>}</pre>
<pre>Status: 406 Not Acceptable</pre>
<pre>{</pre>
<pre>"Error": "Server can only return application/json"</pre>
<pre>}</pre>
<pre>Status: 415 Unsupported Media Type</pre>
<pre>{</pre>
<pre>"Error": "Request is an unsupported media type. Request must be JSON"</pre>
<pre>}</pre>

Assign a Group to a Post *Protected*

PUT /groups/:group_id/posts/:post_id

Request

Path Parameters

Name	Type	Description
group_id	String	ID of the group
post_id	String	ID of the post

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 group exists with this group_id, a post exists with this post_id and the post is not assigned to any group.
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.
Failure	403 Forbidden	The post is already assigned to a group and/or user is not the owner of the post.
Failure	404 Not Found	No group with this group_id exists, and/or no post with this post_id exists.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized
<pre>{ "Error": "Invalid JWT" }</pre>
<pre>{ "Error": "Missing JWT" }</pre>
Status: 403 Forbidden
<pre>{ "Error": "The post is already assigned to a group" }</pre>
<pre>{</pre>

<pre>"Error": "Post is owned by someone else" }</pre>
<pre>Status: 404 Not Found { "Error": "The specified group and post does not exist" } { "Error": "The specified group does not exist" } { "Error": "The specified post does not exist" }</pre>

Comment

- A post cannot be assigned to multiple groups.
- A group can hold multiple posts.
- An already assigned post needs to be removed from another group first before being assigned to another group.

Remove Group From a Post *Protected*

Group is removed from the post. Post does not belong to any group. Group does not have this post.

DELETE /groups/:group_id/posts/:post_id

Request

Path Parameters

Name	Type	Description
group_id	String	ID of the group
post_id	String	ID of the post

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 group exists with this group_id, a post exists with this post_id and the post is not assigned to any group.
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.
Failure	403 Forbidden	The post is not assigned to this group and/or user is not the owner of the post.
Failure	404 Not Found	No group with this group_id exists, and/or no post with this post_id exists.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized { "Error": "Invalid JWT" }
{ "Error": "Missing JWT" }
Status: 403 Forbidden { "Error": "Post not in this group" }
Status: 404 Not Found

```
{  
  "Error": "The specified group and post does not exist"  
}  
  
{  
  "Error": "The specified group does not exist"  
}  
  
{  
  "Error": "The specified post does not exist"  
}
```

Create a Comment *Protected*

Allows you to create a new comment.

POST /posts/:post_id/comments

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
body	String	The body of the comment.	Yes

Request Body Example

Accepted { "body": "I love movies too!", }
Unaccepted (missing at least one attribute) { "random": "I love movies too!" }
Unaccepted (not JSON Format) body: I love movies too!

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	Only "body" will be added to the new entity. Any other attribute will be ignored.
Failure	400 Bad Request	If the request is missing the required attribute, the comment must not be created, and 400 status code must be returned.
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.
Failure	406 Not Acceptable	Response Format is in JSON. If client cannot accept JSON, then comment is not created, and 406 status code is returned.
Failure	415 Unsupported Media Type	Request Format must be in JSON. If client's request format is not in JSON, comment is not created, and 415 status code is returned.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.

- The `self` attribute will contain the live link to the REST resource corresponding to this comment. In other words, this is the URL to get this newly created comment.
- Any other attributes in the request will be ignored and will not be added to the entity.

Success

```
Status: 201 Created
{
  "id": "def123",
  "author": "google-oauth2|103346317843634340680",
  "body": "I love movies too!",
  "creation_date": "06/06/2021 22:59:49",
  "last_modified": "06/06/2021 22:59:49",
  "post_id": "abc123",
  "self": "https://final-suhs.appspot.com/comments/def123"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

```
{
  "Error": "body is not a string"
}
```

```
Status: 401 Unauthorized
```

```
{
  "Error": "Invalid JWT"
}
```

```
{
  "Error": "Missing JWT"
}
```

```
Status: 406 Not Acceptable
```

```
{
  "Error": "Server can only return application/json"
}
```

```
Status: 415 Unsupported Media Type
```

```
{
  "Error": "Request is an unsupported media type. Request must be JSON"
}
```

List all Comments *Protected*

List all the comments.

GET /comments

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.

Response Examples

Success

```
Status: 200 OK
{
  "total_comments": 1,
  "comments": [
    {
      "last_modified": "06/06/2021 23:37:59",
      "author": "google-oauth2|103346317843634340680",
      "body": "I want to watch Ratatouille.",
      "creation_date": "06/06/2021 23:37:59",
      "post_id": "5799384063148032",
      "id": 5236434109726720,
      "self": "https://final-suhs.appspot.com/comments/5236434109726720"
    }
  ]
}
```

Failure

```
Status: 401 Unauthorized
{
  "Error": "Invalid JWT"
}
{
  "Error": "Missing JWT"
}
```


Delete a Comment *Protected*

Allows you to delete a group.

DELETE /comments/:comment_id

Request

Path Parameters

Name	Type	Description
comment_id	String	ID of the comment

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.
Failure	403 Forbidden	If user is not the author of the comment, 403 code is returned.
Failure	404 Not Found	No comment with this comment_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized { "Error": "Invalid JWT" }
{ "Error": "Missing JWT" }
Status: 403 Forbidden { "Error": "Comment is owned by someone else" }
Status: 404 Not Found { "Error": "No comment with this comment_id exists" }

Edit a Subset of a Comment *Protected*

Allows you to edit a subset of a comment's attributes

PATCH /comments/:comment_id

Request

Path Parameters

Name	Type	Description
comment_id	String	ID of the comment

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
body	String	Body of comment.	Yes
Since comment has only one updateable attribute, body of comment is required for edit.			

Request Body Example

<pre>{ "body": "I want to watch Superman." }</pre>
--

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing the required attribute but body is in JSON format with other attributes included, the comment not be created, and 400 status code must be returned.
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.
Failure	403 Forbidden	If user is not the author of the comment, 403 code is returned.
Failure	404 Not Found	No comment with this comment_id exists
Failure	406 Not Acceptable	If the request is an empty body, the comment must not be created, and 406 status code must be returned. Response Format is in JSON. If client cannot accept JSON, then comment is not created, and 406 status code is returned.
Failure	415 Unsupported Media Type	Request Format must be in JSON. If client's request format is not in JSON, comment is not created, and 415 status code is returned.

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this comment. In other words, this is the URL to get this newly created comment.
- Any other attributes in the request will be ignored and will not be added to the entity.

Response Examples

Success

```
Status: 200 OK
{
  "id": "def123",
  "author": "google-oauth2|103346317843634340680",
  "body": "I want to watch Superman.",
  "creation_date": "06/06/2021 22:59:49",
  "last_modified": "06/06/2021 22:59:49",
  "post_id": "abc123",
  "self": "https://final-suhs.appspot.com/comments/def123"
}
```

Failure

```
Status: 400 Bad Request

{
  "Error": "No attribute was given"
}

{
  "Error": "Body is not a string"
}
```

```
Status: 401 Unauthorized
{
  "Error": "Invalid JWT"
}
{
  "Error": "Missing JWT"
}
```

```
Status: 403 Forbidden
{
  "Error": "Comment is owned by someone else"
}
```

```
Status: 404 Not Found

{
  "Error": "No comment with this comment_id exists"
}
```

```
Status: 406 Not Acceptable
```

```
{  
  "Error": "Server can only return application/json"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Request is an unsupported media type. Request must be JSON"  
}
```

Edit all Attributes of a Comment *Protected*

Allows you to edit all attributes of a comment

PUT /comments/:comment_id

Request

Path Parameters

Name	Type	Description
comment_id	String	ID of the comment

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
body	String	Body of comment.	Yes
Since comment has only one updateable attribute, body of comment is required for edit.			

Request Body Example

```
{
  "body": "I want to watch Superman."
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing the required attribute but body is in JSON format with other attributes included, the comment not be created, and 400 status code must be returned.
Failure	401 Unauthorized	If JWT is invalid or missing, 401 code is returned.
Failure	403 Forbidden	If user is not the author of the comment, 403 code is returned.
Failure	404 Not Found	No comment with this comment_id exists
Failure	406 Not Acceptable	If the request is an empty body, the comment must not be created, and 406 status code must be returned. Response Format is in JSON. If client cannot accept JSON, then comment is not created, and 406 status code is returned.

Failure	415 Unsupported Media Type	Request Format must be in JSON. If client's request format is not in JSON, comment is not created, and 415 status code is returned.
---------	----------------------------	---

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this comment. In other words, this is the URL to get this newly created comment.
- Any other attributes in the request will be ignored and will not be added to the entity.

Response Examples

Success

<p>Status: 200 OK</p> <pre>{ "id": "def123", "author": "google-oauth2 103346317843634340680", "body": "I want to watch Superman.", "creation_date": "06/06/2021 22:59:49", "last_modified": "06/06/2021 22:59:49", "post_id": "abc123", "self": "https://final-suhs.appspot.com/comments/def123" }</pre>
--

Failure

<p>Status: 400 Bad Request</p> <pre>{ "Error": "No attribute was given" }</pre> <pre>{ "Error": "Body is not a string" }</pre>
<p>Status: 401 Unauthorized</p> <pre>{ "Error": "Invalid JWT" }</pre> <pre>{ "Error": "Missing JWT" }</pre>
<p>Status: 403 Forbidden</p> <pre>{ "Error": "Comment is owned by someone else" }</pre>
<p>Status: 404 Not Found</p> <pre>{</pre>

<pre>"Error": "No comment with this comment_id exists" }</pre>
<p>Status: 406 Not Acceptable</p> <pre>{ "Error": "Server can only return application/json" }</pre>
<p>Status: 415 Unsupported Media Type</p> <pre>{ "Error": "Request is an unsupported media type. Request must be JSON" }</pre>