

# Portfolio Assignment: API Spec

CS 493: Cloud Application Development

Spring 2020

Oregon State University

Last Update: June 8, 2020. 6:00 pm Pacific

URL: <http://final-tensingn.ue.r.appspot.com>

Login URL: <http://final-tensingn.ue.r.appspot.com/login> (has UI)

Account Creation URL: <http://final-tensingn.ue.r.appspot.com/login/registration> (has UI)

Change log.....	<b>Error! Bookmark not defined.</b>
Data Model/API Description .....	2
Create a Band.....	3
Get a Band.....	5
List all Bands.....	6
Edit a Band - Put.....	8
Edit a Band - Patch .....	10
Delete a Band.....	12
Create Merch .....	13
Get Merch .....	15
List all Merch.....	17
Edit Merch - Put .....	18
Edit Merch - Patch.....	21
Delete Merch .....	18
Merch is Assigned a Band .....	25
Merch is Unassigned from a Band .....	27

## Data Model/API Description

This API has 3 entities: Users, Merch (merchandise such as clothing, music, instruments, etc.), and Bands. Users are the owners of Merch items, which may or may not be associated with a band. For example, a User might have an ACDC Poster that would, of course, be associated with the band ACDC. Users must create the association themselves, however, because the band attribute of a merch item is NULL upon creation. Merch items are protected resources because Merch items are owned by Users. Bands, on the other hand, are public resources and can be accessed by any User.

For Users to make requests, they must login and provide the JWT token ("id\_token" in the response JSON object) given to them upon their login for any request involving a protected resource (i.e. Merch requests). The userid attribute of the user (see below) is the "sub" value from the JWT token.

### User Attributes

Name	Type	Description	Required?
id	String	ID as stored in the database.	N/A
email	String	Email address of the user.	Yes
userid	String	Unique ID assigned to user based on the sub value of their JWT token.	N/A

### Merch Attributes

Name	Type	Description	Required?
description	String	A description of the merch item. E.g., Chris Brown CD, Red Hot Chili Peppers Poster, etc.	Yes
type	String	The type of item the merch is. E.g., music, clothing, etc.	No
condition	String	The condition of the merch item. E.g., new, used, etc.	No
band	Object	ID and link to the band associated with the merch item. Set to NULL upon creation.	N/A
user	String	The user who created the merch item.	N/A
id	String	ID of the merch item.	N/A
self	String	URL of the location of the resource	N/A

### Band Attributes

Name	Type	Description	Required?
name	String	The name of the band.	Yes
genre	String	The genre of the band. E.g., Heavy Metal, Hip Hop, etc.	No
country	String	The band's country of origin.	No
merch	Array of objects	IDs and links to each of the merch items associated with the band. Empty array upon creation.	N/A
id	String	ID of the merch item.	N/A
self	String	URL of the location of the resource	N/A

## Create a Band

Allows you to create a new band.

POST /bands

### Request

Request Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the band.	Yes
genre	String	The genre of the band. E.g., Heavy Metal, Hip Hop, etc.	No
country	String	The band's country of origin.	No

Request Body Example

```
{
  "name": "Metallica",
  "genre": "Heavy Metal",
  "country": "United States of America"
}
```

### Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing the required attribute, the band must not be created, and 400 status code must be returned.

Response Examples

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

### Success

```
Status: 201 Created
{
```

```
{
  "id": "abc123",
  "name": "Metallica",
  "genre": "Heavy Metal",
  "country": "United States of America",
  "merch": [],
  "self": "http://final-tensingn.ue.r.appspot.com/bands/abc123"
}
```

#### *Failure*

Status: 400 Bad Request

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

## Get a Band

Allows you to get an existing band.

GET /bands/:band_id
---------------------

### Request

#### Request Parameters

Name	Type	Description	Required?
band_id	String	ID of the band	Yes

#### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No band with this ID exists.

#### Response Examples

##### Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Metallica",
  "genre": "Heavy Metal",
  "country": "United States of America",
  "merch": [
    {
      "id": "def456",
      "self": "http://final-tensingn.ue.r.appspot.com/merch/def456"
    }
  ],
  "self": "http://final-tensingn.ue.r.appspot.com/bands/abc123"
}
```

##### Failure

```
Status: 404 Not Found

{
  "Error": "No band with this ID exists"
}
```

## List all Bands

List all the bands.

GET /bands
------------

### Request

Request 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

```
[
  {
    "id": "abc123",
    "name": "Metallica",
    "genre": "Heavy Metal",
    "country": "United States of America",
    "merch": [
      {
        "id": "def456",
        "self": "http://final-tensingn.ue.r.appspot.com/merch/def456"
      }
    ],
    "self": "http://final-tensingn.ue.r.appspot.com/bands/abc123"
  },
  {
    "id": "ghi789",
    "name": "Arctic Monkeys",
    "genre": "Indie Rock",
    "country": "England",
    "merch": [],
    "self": "http://final-tensingn.ue.r.appspot.com/bands/ghi789"
  },
  {
    "id": "xyz999",
    "name": "Tame Impala",
    "genre": "Neo-Psychedelia",
```

```
"country": "Australia",
"merch": [
  {
    "id": "abc456",
    "self": "http://final-tensingn.ue.r.appspot.com/merch/abc456"
  },
  "self": "http://final-tensingn.ue.r.appspot.com/bands/xyz999"
]
```

## Edit a Band - Put

Allows you to edit a band. Note that all unspecified attributes will be updated to NULL.

PUT /bands/:band\_id

### Request

#### Request Parameters

Name	Type	Description	Required?
band_id	String	ID of the band	Yes

#### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the band.	Yes
genre	String	The genre of the band. E.g., Heavy Metal, Hip Hop, etc.	Yes
country	String	The band's country of origin.	Yes

#### Request Body Example

```
{
  "name": "Metallica",
  "genre": "Thrash Metal",
  "country": "United States of America"
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If attempting to change ID.
Failure	404 Not Found	No band with this ID exists

#### Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this band. You must not store this attribute in Datastore.

#### Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Metallica",
  "genre": "Thrash Metal",
```



```
"country": "United States of America",  
"merch": [],  
"self": "http://final-tensingn.ue.r.appspot.com/bands/abc123"  
}
```

#### *Failure*

Status: 400 Bad Request

```
{  
"Error": "Cannot change id"  
}
```

Status: 404 Not Found

```
{  
"Error": "No band with this ID exists."  
}
```

## Edit a Band - Patch

Allows you to edit a band without having to specify all the band's attributes.

PATCH /bands/:band_id
-----------------------

### Request

#### Request Parameters

Name	Type	Description	Required?
band_id	String	ID of the band	Yes

#### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the band.	No
genre	String	The genre of the band. E.g., Heavy Metal, Hip Hop, etc.	No
country	String	The band's country of origin.	No

#### Request Body Example

```
{
  "genre": "Thrash Metal"
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If attempting to change ID.
Failure	404 Not Found	No band with this ID exists

#### Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this band. You must not store this attribute in Datastore.

#### Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Metallica",
  "genre": "Thrash Metal",
  "country": "United States of America",
  "merch": [
```

```
{
  {
    "id": "def456",
    "self": "http://final-tensingn.ue.r.appspot.com/merch/def456"
  },
  "self": "http://final-tensingn.ue.r.appspot.com/bands/abc123"
}
```

#### *Failure*

Status: 400 Bad Request

```
{
  "Error": "Cannot change id"
}
```

Status: 404 Not Found

```
{
  "Error": "No band with this ID exists."
}
```

## Delete a Band

Allows you to delete a band. Note that if the band is currently associated with any merch items, deleting the band makes all the merch items no longer associated with the band.

DELETE /bands/:band_id
------------------------

### Request

Request Parameters

None

Request Body

None

### Response

No body

Response Body Format

Success: No body

Failure: JSON

### Response Statuses

Outcome	Status Code	Notes
Success	204 No Type	
Failure	404 Not Found	No band with this ID exists

### Response Examples

#### Success

Status: 204 No Type
---------------------

#### Failure

Status: 404 Not Found  <pre>{   "Error": "No band with this ID exists" }</pre>
--

## Create Merch

Allows you to create a new merch.

POST /merch
-------------

### Request

#### Request Authorization

Must include a valid JWT token as a Bearer Token.

#### Request Parameters

None

#### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Type	Description	Required?
description	String	A description of the merch item. E.g., Chris Brown CD, Red Hot Chili Peppers Poster, etc.	Yes
type	String	The type of item the merch is. E.g., music, clothing, etc.	No
condition	String	The condition of the merch item. E.g., new, used, etc.	No

#### Request Body Example

```
{
  "description": "Black Keys CD",
  "type": "music",
  "condition": "new"
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing the required attribute, the merch must not be created, and 400 status code must be returned.
Failure	401 Unauthorized	If the user making the request provides a bad JWT token, the merch must not be created, and 401 status code must be returned.

## Response Examples

- The value of the attribute `band` is the band associated with this merch item. The attribute `band` starts out null when the merch is created.
- The value of the attribute `self` is a live link to the REST resource corresponding to this merch. In other words, this is the URL to get this newly created merch. You must not store this attribute in Datastore

### Success

Status: 201 Created

```
{
  "id": "123abc",
  "description": "Black Keys CD",
  "type": "music",
  "condition": "new",
  "band": null,
  "self": "http://final-tensingn.ue.r.appspot.com/merch/123abc"
}
```

### Failure

Status: 400 Bad Request

```
{
  "Error": "The request object is missing a required attribute."
}
```

### Failure

Status: 401 Unauthorized

```
{
  "Error": "Invalid Token."
}
```

## Get Merch

Allows you to get an existing merch.

GET /merch/:merch_id
----------------------

### Request

#### Request Authorization

Must include a valid JWT token as a Bearer Token. The token must also prove that the requester is allowed to access the resource.

#### Request Header

Accept must be set to "application/json."

#### Request Parameters

Name	Type	Description	Required?
merch_id	String	ID of the merch	Yes

### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Not Authorized	If the user making the request provides a bad JWT token, the merch must not be accessed, and 401 status code must be returned.
Failure	403 Forbidden	The token provide is a valid token, but the user is not allowed to access the resource.
Failure	404 Not Found	No merch with this ID exists.
Failure	406 Not Acceptable	The Accept value in the Request header is not application/JSON.

### Response Examples

#### Success

Status: 200 OK
<pre>{   "id": "123abc",   "description": "Black Keys CD",   "type": "music",   "condition": "new",   "band": {     "id": rst838,     "self": "http://appspot.com/bands/rst838"   }   "self": "http://final-tensingn.ue.r.appspot.com/merch/123abc"</pre>

```
}
```

*Failure*

Status: 401 Not Authorized

```
{  
  "Error": "Invalid Token."  
}
```

*Failure*

Status: 403 Forbidden

```
{  
  "Error": "Forbidden. You do not have access to this merch item."  
}
```

*Failure*

Status: 404 Not Found

```
{  
  "Error": "No merch with this ID exists"  
}
```

*Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Not Acceptable."  
}
```



## List all Merch for a User

List all the merch for a user.

GET /merch
------------

### Request

#### Request Authorization

Must include a valid JWT token as a Bearer Token.

#### Request Parameters

None

#### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Not Authorized	If the user making the request provides a bad JWT token, no list can be accessed, and 401 status code must be returned.

#### Response Examples

##### Success

Status: 200 OK

```
[
{
  "id": "123abc",
  "description": "Black Keys CD",
  "type": "music",
  "condition": "new",
  "user": "test2@test.com",
  "band": {
    "id": "abc123",
    "self": "http://appspot.com/bands/abc123"
  },
  "self": "http://final-tensingn.ue.r.appspot.com/merch/123abc"
},
{
  "id": "456abc",
  "description": "Black Keys CD",
  "type": "type8",
  "condition": "new",
  "user": "test@test.com",
  "band": {
```

```
{
  "id": abc456,
  "self": "http://appspot.com/bands/abc456"
}
{"self": "http://final-tensingn.ue.r.appspot.com/merch/456abc"
}]
```

#### Failure

```
Status: 401 Not Authorized

{
  "Error": "Invalid Token."
}
```

## Edit a Merch - Put

Allows you to edit a merch item. Note that all unspecified attributes will be updated to NULL.

PUT /merch/:merch\_id

#### Request

##### Request Parameters

Name	Type	Description	Required?
merch_id	String	ID of the merch item	Yes

#### Request Body

Required

#### Request Body Format

JSON

##### Request JSON Attributes

Name	Type	Description	Required?
description	String	A description of the merch item. E.g., Chris Brown CD, Red Hot Chili Peppers Poster, etc.	Yes
type	String	The type of item the merch is. E.g., music, clothing, etc.	Yes
condition	String	The condition of the merch item. E.g., new, used, etc.	Yes

#### Request Body Example

```
{
  "description": "Black Keys CD",
  "type": "music",
  "condition": "used"
}
```

#### Response

##### Response Body Format

JSON

## Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If attempting to change ID.
Failure	401 Not Authorized	If the user making the request provides a bad JWT token, the merch must not be accessed, and 401 status code must be returned.
Failure	403 Forbidden	The token provide is a valid token, but the user is not allowed to access the resource.
Failure	404 Not Found	No band with this ID exists

## Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this band. You must not store this attribute in Datastore.

### Success

Status: 200 OK

```
{
  "id": "123abc",
  "description": "Black Keys CD",
  "type": "music",
  "condition": "used",
  "band": null,
  "user": "test@test.com",
  "self": "http://final-tensingn.ue.r.appspot.com/merch/123abc"
}
```

### Failure

Status: 400 Bad Request

```
{
  "Error": "Cannot change id"
}
```

### Failure

Status: 401 Not Authorized

```
{
  "Error": "Invalid Token."
}
```

### Failure

Status: 403 Forbidden

```
{
  "Error": "Forbidden. You do not have access to this merch item."
}
```

### Failure

Status: 404 Not Found

```
{  
  "Error": "No merch with this ID exists."  
}
```

## Edit a Merch - Patch

Allows you to edit a merch item without having to specify all attributes.

```
PUT /merch/:merch_id
```

### Request

#### Request Parameters

Name	Type	Description	Required?
merch_id	String	ID of the merch item	Yes

#### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Type	Description	Required?
description	String	A description of the merch item. E.g., Chris Brown CD, Red Hot Chili Peppers Poster, etc.	No
type	String	The type of item the merch is. E.g., music, clothing, etc.	No
condition	String	The condition of the merch item. E.g., new, used, etc.	No

#### Request Body Example

```
{
  "condition": "used"
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If attempting to change ID.
Failure	401 Not Authorized	If the user making the request provides a bad JWT token, the merch must not be accessed, and 401 status code must be returned.
Failure	403 Forbidden	The token provide is a valid token, but the user is not allowed to access the resource.
Failure	404 Not Found	No band with this ID exists

#### Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this band. You must not store this attribute in Datastore.

#### Success

```
Status: 200 OK
```

```
{
  "id": "123abc",
  "description": "Black Keys CD",
  "type": "music",
  "condition": "used",
  "band": null,
  "user": "test@test.com",
  "self": "http://final-tensingn.ue.r.appspot.com/merch/123abc"
}
```

*Failure*

Status: 400 Bad Request

```
{
  "Error": "Cannot change id"
}
```

*Failure*

Status: 401 Not Authorized

```
{
  "Error": "Invalid Token."
}
```

*Failure*

Status: 403 Forbidden

```
{
  "Error": "Forbidden. You do not have access to this merch item."
}
```

*Failure*

Status: 404 Not Found

```
{
  "Error": "No merch with this ID exists."
}
```

## Delete Merch

Allows you to delete merch. If the merch being deleted is currently associated with a band, the merch is removed from the bands list of merch.

DELETE /merch/:merch_id
-------------------------

### Request

#### Request Authorization

Must include a valid JWT token as a Bearer Token. The token must also prove that the requester is allowed to access the resource.

#### Request Parameters

Name	Type	Description	Required?
merch_id	String	ID of the merch	Yes

#### Request Body

None

### Response

No body

#### Response Body Format

Success: No body

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Type	
Failure	401 Not Authorized	If the user making the request provides a bad JWT token, the resource must not be deleted, and 401 status code must be returned.
Failure	403 Forbidden	The token provided is a valid token, but the user is not allowed to delete the resource.
Failure	404 Not Found	No merch with this merch_id exists

#### Response Examples

##### Success

Status: 204 No Type
---------------------

##### Failure

Status: 401 Not Authorized
<pre>{   "Error": "Invalid Token." }</pre>

##### Failure

Status: 403 Forbidden
<pre>{</pre>

```
"Error": "Forbidden. You do not have access to this merch item."
}
```

#### *Failure*

Status: 404 Not Found

```
{
  "Error": "No merch with this ID exists"
}
```



## Merch is Assigned a Band

Merch is now associated with a band.

PUT /merch/:merch_id/bands/:band_id
-------------------------------------

### Request

#### Request Authorization

Must include a valid JWT token as a Bearer Token. The token must also prove that the requester is allowed to access the resource.

#### Request Parameters

Name	Type	Description	Required?
merch_id	String	ID of the merch	Yes
band_id	String	ID of the band	Yes

#### Request Body

None

### Response

No body

#### Response Body Format

Success: No body

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Type	Succeeds only if a band exists with this band_id, Merch exists with this merch_id and the merch is not already associated with a band.
Failure	401 Not Authorized	If the user making the request provides a bad JWT token, the resource must not be deleted, and 401 status code must be returned.
Failure	403 Forbidden	This merch is already associated with a band.
Failure	403 Forbidden	The token provided is a valid token, but the user is not allowed to access the resource.
Failure	404 Not Found	No band with this band_id exists, and/or no merch with this merch_id exists.

#### Response Examples

##### Success

Status: 204 No Type
---------------------

##### Failure

Status: 401 Not Authorized
<pre>{   "Error": "Invalid Token." }</pre>

#### Failure

Status: 403 Forbidden

```
{  
  "Error": "Forbidden. This merch item is already associated with a band."  
}
```

#### Failure

Status: 403 Forbidden

```
{  
  "Error": "Forbidden. You do not have access to this merch item.""  
}
```

Status: 404 Not Found

```
{  
  "Error": "The specified band and/or merch don't exist"  
}
```

#### Comment

- A band can be associated with multiple merch items.
- Merch can only be associated with one band.

## Merch is Unassigned from a Band

Band is no longer associated with the merch item.

DELETE /merch/:merch_id/bands/:band_id/
---

### Request

#### Request Authorization

Must include a valid JWT token as a Bearer Token. The token must also prove that the requester is allowed to access the resource.

#### Request Parameters

Name	Type	Description	Required?
merch_id	String	ID of the merch	Yes
band_id	String	ID of the band	Yes

### Request Body

None

### Response

No body

#### Response Body Format

Success: No body

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Type	Succeeds only if a band exists with this band_id, Merch exists with this merch_id and this merch is on this band.
Failure	401 Not Authorized	If the user making the request provides a bad JWT token, the resource must not be deleted, and 401 status code must be returned.
Failure	403 Forbidden	The token provided is a valid token, but the user is not allowed to access the resource.
Failure	404 Not Found	No band with this band_id is associated with the merch with this merch_id.

#### Response Examples

##### Success

Status: 204 No Type
---------------------

##### Failure

Status: 401 Not Authorized
<pre>{   "Error": "Invalid Token." }</pre>

##### Failure

Status: 403 Forbidden
-----------------------

```
{  
  "Error": "Forbidden. You do not have access to this merch item."  
}
```

#### *Failure*

Status: 404 Not Found

```
{  
  "Error": "The band with this band_id does not carry the merch with this merch_id"  
}
```