Portfolio Project: API Spec

Deployed Application / Login Landing URL: https://final-eckermae.wl.r.appspot.com

Erin Eckerman

CS 493: Cloud Application Development

Fall 2022

Oregon State University

Last Update: Nov 30, 2022

Data Model

The app stores three kinds of entities in Datastore, Users, Fabrics, and Patterns. There is no endpoint for user record creation - user records are created automatically at time of first login.

Users

Property	Data Type	Notes	Required	Valid Values
uid	Integer	Unique ID of user - is the sub of the user's OAuth JWT	Y	Any positive integer
given_name	String	First name	Y	Any string
family_name	String	Last name	Y	Any string
join_date	Datetime	Date and time user's record was created	Y	YYYY-MM-DD HH:MM:SS.MS+TZ e.g. "2022-11-28 12:26:33.523124+00:00"
patterns	List of Integers	IDs of Patterns assigned to the user	N	Any ids in Patterns

Fabrics

Property	Data Type	Notes	Required	Valid Values
id	Integer	The id of the fabric - automatically generated by Datastore	Y	Any positive integer
substrate	String	Type of fabric	Y	Any string representing a substrate e.g. "cotton"
color	String	Color of fabric	Y	Any string representing a color e.g. "black"
yards	Integer	Number of yards available for fabric - must be > 0	Y	Any integer > 0
patterns	List of Integers	ID of pattern(s) to which fabric is assigned - can be null	N	Any id in Patterns

Patterns

Property	Data Type	Notes	Required	Valid Values
id	Integer	The id of the pattern - automatically generated by Datastore	Y	Any positive integer
name	String	Name of the pattern	Y	Any string representing a pattern name

maker	String	ID of user to which pattern is assigned - can be null	N	Any id in Users
fabric	String	ID of fabric that is assigned to pattern - can be null	N	Any id in Fabrics
garment	String	Type of pattern	Y	Any string representing a type of garment e.g. “dress”
company	String	Name of company that produces the pattern	Y	Any string representing a pattern company e.g. “Closet Core”

Relationship Between Fabrics and Patterns

There is a 1:Many relationship between Fabrics and Patterns - i.e. the same fabric record may be associated with multiple pattern records but patterns can only have a single fabric associated with it. It is possible for fabrics to not be associated with any patterns and for patterns to not have any fabrics assigned to it. Any updates to the “fabric” value of a Pattern record must supply the id of the fabric and any updates to the “pattern” value of a Fabric record must supply the id of the pattern. Any updates to the relationship between patterns and fabrics or deletions of either a Pattern or Fabric record will be cascaded to any associated records.

User Entity Modeling

The unique identifier for a User is the uid - this is generated at the time of User record creation and is derived from the “sub” within the user’s JWT provided by Google OAuth. There is a 1:Many relationship between Users and Patterns - i.e. the same user record may be associated with multiple pattern records but patterns can only have a single user associated with it. If a Pattern record is going to be assigned to a user (a.k.a. “maker”), the uid must be provided as part of the update to the Pattern record - any endpoints that update the “maker” of a Pattern record will also update the “patterns” value on the corresponding User record and any deletions of either a Pattern or User record will be cascaded to associated User or Pattern records.

Create a Fabric

Allows you to create a new fabric.

```
POST /fabrics
```

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
substrate	The type of fabric	Yes
color	The color of the fabric	Yes
yards	Number of yards of the fabric	Yes

Request Body Example

```
{
  "substrate": "cotton",
   "color": "blue",
  "yards": 3
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes

Response Examples

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

Success

```
Status: 201 Created

{
  "color": "blue",
   "id": "5925479739555840",
   "patterns": [],
   "self": "https://final-eckermae.wl.r.appspot.com/5925479739555840",
   "substrate": "cotton",
   "yards": 3
}
```

Failure

```
Status: 400 Bad Request

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

List All Fabric

Returns a paginated list of Fabric records - 5 records at a time using limit and offset variables within the url - and includes a “next” url in every response except at the end of the list of records.

```
GET /fabrics
```

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
{
  "fabrics": [
    {
      "substrate": "silk",
      "yards": 3,
      "patterns": [
        null
      ],
      "color": "rainbow",
      "pattern": null,
      "self": "https://final-eckermae.wl.r.appspot.com/fabrics/4626141671325696",
      "id": "4626141671325696"
    },
    {
      "yards": 1,
      "color": "white",
      "substrate": "bamboo",
      "self": "https://final-eckermae.wl.r.appspot.com/fabrics/4670788124803072",
      "id": "4670788124803072",
      "patterns": []
    },
    {
      "yards": 2,
      "color": "blue",
      "substrate": "cotton",
      "self": "https://final-eckermae.wl.r.appspot.com/fabrics/4792952966610944",
      "id": "4792952966610944",
      "patterns": []
    },
    {
      "yards": 2,
      "color": "red",
      "substrate": "cotton",
      "self": "https://final-eckermae.wl.r.appspot.com/fabrics/4806027115495424",
      "id": "4806027115495424",
      "patterns": []
    },
    {
      "yards": 1,
      "color": "white",
      "substrate": "bamboo",
      "self": "https://final-eckermae.wl.r.appspot.com/fabrics/4818367227625472",
      "id": "4818367227625472",
      "patterns": []
    }
  ],
  "next": "https://final-eckermae.wl.r.appspot.com/fabrics?limit=5&offset=5",
  "total_items": 45
}
```

Get a Fabric

Allows you to get an existing fabric.

```
GET /fabrics/:fabric_id
```

Request

Path Parameters

Name	Description
fabric_id	ID of the fabric

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No fabric with this fabric_id exists

Response Examples

Success

```
Status: 200 OK
{
  "yards": 3,
  "color": "blue",
  "substrate": "cotton",
  "patterns": [],
  "id": "5351024105619456",
  "self": "https://final-eckermae.wl.r.appspot.com/fabrics/5351024105619456"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No fabric with this fabric_id exists"
}
```

Delete a Fabric

Allows you to delete a fabric. Note that if the fabric is assigned to a pattern, the “fabric” of the pattern will be changed to None.

```
DELETE /fabrics/:fabric_id
```

Request

Path Parameters

Name	Description
fabric_id	ID of the fabric

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 fabric with this fabric_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

{
"Error":  "No fabric with this fabric_id exists"
}
```

Update a Fabric

Updates an existing fabric and only updates/overwrites attributes included in the request body.

```
PUT or PATCH  /fabrics/:fabric_id
```

Request

Path Parameters

Name	Description
fabric_id	ID of the fabric

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
substrate	The type of fabric	No
color	The color of the fabric	No
yards	Number of yards of the fabric	No

Request Body Example

```
{
  "yards": 2
}
```

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No fabric with this fabric_id exists

Response Examples

Success

```
Status: 200 OK
{
  "yards": 2,
  "color": "blue",
  "substrate": "cotton",
  "patterns": [],
  "id": "5351024105619456",
  "self": "https://final-eckermae.wl.r.appspot.com/fabrics/5351024105619456"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No fabric with this fabric_id exists"
}
```

Create a Pattern

Allows you to create a new pattern. A bearer token in the Authorization header of the request is required. After the user’s token is verified, that user’s id will be assigned as the “maker” attribute of the pattern.

```
POST /pattern
```

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	**Description**	**Required?**
name	The name of the pattern	Yes
garment	The type of garment	Yes
company	The name of the pattern company	Yes

Request Body Example

```
{
   "name": "Agave",
   "garment": "skirt",
   "company": "Deer & Doe"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes
Failure	401 Unauthorized	

Response Examples

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

Success

```
Status: 201 Created

{
   "company": "Deer & Doe",
   "fabric": null,
    "garment": "skirt",
    "id": "6335358803378176",
    "maker": "109740306570073680111",
    "name": "Agave",
    "self": "https://final-eckermae.wl.r.appspot.com/patterns/6335358803378176"
}
```

Failure

```
Status: 400 Bad Request

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

```
Status: 401 Unauthorized
```

List All Patterns

Returns a paginated list of pattern records belonging to logged in user - 5 records at a time using limit and offset variables within the url - and includes a “next” url in every response except at the end of the list of records.

```
GET /patterns
```

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	

Response Examples

Success

```
Status: 200 OK
{
  "patterns": [
    {
      "company": "Closet Core",
      "garment": "dress",
      "fabric": null,
      "maker": "109740306570073680111",
      "name": "Jenny",
      "self": "https://final-eckermae.wl.r.appspot.com/patterns/4532102422855680",
      "id": "4532102422855680"
    },
    {
      "company": "Pattern Emporium",
      "garment": "shirt",
      "maker": "109740306570073680111",
      "fabric": null,
      "name": "Long Goodbye",
      "self": "https://final-eckermae.wl.r.appspot.com/patterns/4619637882880000",
      "id": "4619637882880000"
    },
    {
      "company": "Paper Theory Patterns",
      "garment": "jumpsuit",
      "maker": "109740306570073680111",
      "name": "Zadie",
      "self": "https://final-eckermae.wl.r.appspot.com/patterns/4828657734385664",
      "id": "4828657734385664",
      "fabric": null
    },
    {
      "company": "Pattern Emporium",
      "garment": "shirt",
      "maker": "109740306570073680111",
      "name": "Hello Gorgeous",
      "self": "https://final-eckermae.wl.r.appspot.com/patterns/4862202200719360",
      "id": "4862202200719360",
      "fabric": null
    },
    {
      "company": "Pattern Emporium",
      "garment": "shirt",
      "maker": "109740306570073680111",
      "name": "Hello Gorgeous",
      "self": "https://final-eckermae.wl.r.appspot.com/patterns/4882541421002752",
      "id": "4882541421002752",
      "fabric": null
    }
  ],
  "next": "https://final-eckermae.wl.r.appspot.com/patterns?limit=5&offset=5",
  "total_items": 21
}
```

Failure

```
Status: 401 Unauthorized
```

Get a Pattern

Allows user to get an existing pattern if they are the “maker” listed on the pattern record.

```
GET /patterns/:pattern_id
```

Request

Path Parameters

Name	Description
pattern_id	ID of the pattern

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	
Failure	404 Not Found	No pattern with this pattern_id exists
Failure	403 Forbidden	ID in “maker” field does not match user’s ID

Response Examples

Success

```
Status: 200 OK
{
 "company": "Closet Core",
  "garment": "overalls",
  "maker": "109740306570073680111",
  "name": "Jenny",
  "fabric": [],
  "id": "6551438742257664",
  "self": "https://final-eckermae.wl.r.appspot.com/patterns/6551438742257664"
}
```

Failure

```
Status: 401 Unauthorized
```

```
Status: 404 Not Found
{
"Error":  "No pattern with this pattern_id exists"
}
```

```
Status: 403 Forbidden
{
"Error":  "You are not authorized to access this resource"
}
```

Delete a Pattern

Allows user to delete an existing pattern if they are the "maker" listed on the pattern record. If there is a fabric listed within the pattern record, the pattern id will be removed from that fabric record's patterns field.

```
DELETE /patterns/:pattern_id
```

Request

Path Parameters

Name	Description
pattern_id	ID of the pattern

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	
Failure	403 Forbidden	
Failure	404 Not Found	No fabric with this fabric_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized
```

```
Status: 403 Forbidden
{
"Error":  "You are not authorized to access this resource"
}
```

```
Status: 404 Not Found
{
"Error":  "No pattern with this pattern_id exists"
}
```

Update a Pattern

Allows user to update an existing pattern if they are the “maker” listed on the pattern record.

```
PUT or PATCH  /patterns/:pattern_id
```

Request

Path Parameters

Name	Description
pattern_id	ID of the pattern

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the pattern	Yes
garment	The type of garment	Yes
company	The name of the pattern company	Yes

Request Body Example

```
{
   "garment": "dress"
}
```

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	
Failure	403 Forbidden	
Failure	404 Not Found	No pattern with this pattern_id exists

Response Examples

Success

```
Status: 200 OK
{
 "garment": "dress",
  "fabric": [],
  "company": "Closet Core",
  "maker": "109740306570073680111",
  "name": "Jenny",
  "id": 4677581555105792,
  "self": "https://final-eckermae.wl.r.appspot.com/patterns/4677581555105792"
}
```

Failure

```
Status: 401 Unauthorized
```

```
Status: 403 Forbidden
{
"Error":  "You are not authorized to access this resource"
}
```

```
Status: 404 Not Found
{
"Error":  "No pattern with this pattern_id exists"
}
```

Add Fabric to a Pattern

Allows user to add a fabric id to an existing pattern if they are the “maker” listed on the pattern record.

```
PUT or PATCH  /patterns/:pattern_id/fabrics/:fabric_id
```

Request

Path Parameters

Name	Description
pattern_id	ID of the pattern
fabric_id	ID of the fabric

Request Body

None

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	
Failure	403 Forbidden	
Failure	404 Not Found	No pattern with this pattern_id exists

Response Examples

Success

```
Status: 200 OK
{
  "company": "Closet Core",
  "garment": "dress",
  "maker": "109740306570073680111",
  "name": "Jenny",
  "fabric": {
    "yards": 2,
    "color": "blue",
    "substrate": "cotton",
    "patterns": [
      "6289916304556032"
    ]
  },
  "self": "https://final-eckermae.wl.r.appspot.com/patterns/6289916304556032",
  "id": 6289916304556032
}
```

Failure

```
Status: 401 Unauthorized
```

Status: 403 Forbidden (does not belong to user)

```
{
"Error":  "You are not authorized to access this resource"
}
```

Status: 403 Forbidden (there is already a fabric assigned to the pattern)

```
{
"Error":  "There is already a fabric assigned to this pattern"
}
```

Status: 404 Not Found

```
{
"Error":  "No pattern with this pattern_id exists or no fabric with this fabric_id
exists"
}
```

Remove Fabric from a Pattern

Allows user to remove a fabric id from an existing pattern if they are the “maker” listed on the pattern record.

```
DELETE  /patterns/:pattern_id/fabrics/:fabric_id
```

Request

Path Parameters

Name	Description
pattern_id	ID of the pattern
fabric_id	ID of the fabric

Request Body

None

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	
Failure	403 Forbidden	
Failure	404 Not Found	No pattern with this pattern_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized
```

```
Status: 403 Forbidden (does not belong to user)
{
"Error":  "You are not authorized to access this resource"
}
```

```
Status: 403 Forbidden (there is no fabric id in pattern record)
{
"Error":  "There is no fabric to remove from this pattern"
}
```

```
Status: 403 Forbidden (fabric id in pattern record does not match given fabric id)
{
"Error":  "This fabric_id is not associated with this pattern"
}
```

```
Status: 404 Not Found
{
"Error":  "No pattern with this pattern_id exists or no fabric with this fabric_id exists"
}
```

List All Users

Returns a list of 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
{
[
  {
    "family_name": "Eckerman",
    "given_name": "Erin",
    "join_date": "2022-11-29 12:30:59.685072+00:00",
    "uid": "112619984909112801964"
  },
  {
    "family_name": "Gheorghe",
    "given_name": "Bogdan",
    "join_date": "2022-11-28 12:26:33.523000+00:00",
    "uid": "109740306570073680111"
  }
]
}
```