Final Project: API Spec

CS 493: Cloud Application Development | William Lambeth | Spring 2023

App URL: **https://final-493-wfl.wl.r.appspot.com/**

Data Model 2

Create user: GET / 3

View all users: GET /users 4

Create a Pizza: POST /pizzas 5

View a Pizza: GET /pizzas/:id 7

Update a Pizza (partial): PATCH /pizzas/:id 13

Create a Topping: POST /toppings 16

View all toppings: GET /toppings 18

Update a topping: PUT /toppings/:id 20

Update a topping (partial): PATCH /toppings/:id 22

Add topping to pizza: PUT /pizzas/:pid/topping/:tid 25

Remove topping from pizza: DELETE /pizzas/:pid/topping/:tid 27

Data Model

This app stores three types of entities: **users**, **pizzas**, and **toppings**.

User

Property	**Data Type**	**Notes**
ID	string	derived from the 'sub' value of a Google OAuth JWT

Pizza

Property	**Data Type**	**Notes**
ID	int	auto-generated, unique, immutable
owner	string	auto-generated; "ID" value of creating User
name	string	cannot be empty
size	int	cannot be zero/negative
style	string	
self	string	auto-generated; canonical URL of this entity
topping	string	"name" value of associated topping entity (if one exists)
topping_id	int	"ID" value of associated topping entity (if one exists)

Topping

Property	**Data Type**	**Notes**
ID	int	auto-generated, unique, immutable
name	string	cannot be empty
spiciness	int	
tastiness	int	
traditional	bool	
self	string	auto-generated; canonical URL of this entity

Create user: GET /

Note: This path involves clicking a button on an HTML page, then being redirected to Google OAuth to log in and generate a JWT.

(Any error or failure to authenticate will lead to a Google-generated error response, which is not in the scope of this document.)

Request

Path	GET /

Response

Format: HTML

Outcome	**Status code**	**Notes**
Success; JWT generated	200	User will be redirected to **/auth_info** on successful authentication. This page cannot be accessed directly.

View all users: GET /users

Request

Path	GET /users
Accept	application/json

Response

Format: JSON

Outcome	**Status**	**Notes**
Success	200	Returns list of all user IDs in database, in JSON format
Failure	406	Returned if "Accept: application/json" not in req. headers

Body Examples:

```
200 OK
{
"id": "118077088807767242950",
"id": "432081239012309123103"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

Create a Pizza: POST /pizzas

Request

Path	POST /pizzas
Accept	application/json
Authorization	Valid JWT as bearer token
Body req.?	Y
Body Format	JSON

Body Attributes:

Name	**Description**	**Required?**
name	String providing a name for your pizza; no uniqueness constraint	Y
size	Integer describing pizza's size (diameter in inches)	Y
style	String describing the style of your pizza (generally refers to crust)	Y

Body Example:

```
{
"name": "My New 'Za",
"size": 21,
"style": "Sicilian"
}
```

Response

Format: JSON

Outcome	**Status**	**Notes**
Success	201	Creates and returns DB representation, including autogenerated ID.
Failure	400	Returned if data provided is invalid (i.e. type error, missing fields)
Failure	401	Returned if no valid JWT is provided in request headers
Failure	403	Returned if JWT provided is not registered with the app
Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
201 CREATED
{
"id": 5,
"owner": "118077088807767242950",
"name": "My New 'Za",
"size": 21,
"style": "Sicilian",
"topping": "",
"self": "https://final-493-wfl.wl.r.appspot.com/pizzas/5"
}

400 BAD REQUEST
{
"Error: Invalid data provided"
}

401 UNAUTHORIZED
{
"Error: must provide a valid JWT in header"
}

403 FORBIDDEN
{
"Error: JWT provided is not for a registered user"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

View a Pizza: GET /pizzas/:id

Request

Path	GET /pizzas/:id
Accept	application/json
Authorization	Valid JWT as bearer token

Response

Format: JSON

Outcome	**Status**	**Notes**
Success	200	Returns full DB representation of pizza.
Failure	401	Returned if no valid JWT is provided in request headers
Failure	403	Returned if JWT provided is not the owner of the specified pizza
Failure	404	Returned if ID provided does not match a pizza in the DB
Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
200 OK
{
"id": 5,
"owner": "118077088807767242950",
"name": "My New 'Za",
"size": 21,
"style": "Sicilian",
"topping": "",
"self": "https://final-493-wfl.wl.r.appspot.com/pizzas/5"
}

401 UNAUTHORIZED
{
"Error: must provide a valid JWT in header"
}

403 FORBIDDEN
```

```
{
"Error: Not authorized to access this pizza"
}

404 NOT FOUND
{
"Error: Pizza ID Not Found"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

View all pizzas for this user: GET /pizzas

Request

Path	GET /pizzas
Accept	application/json
Authorization	Valid JWT as bearer token

Response

Format: JSON

Outcome	Status	Notes
Success	200	Returns list of all pizzas associated with user who owns JWT. Paginated if necessary; also contains total count of user's pizzas.
Failure	401	Returned if no valid JWT is provided in request headers
Failure	403	Returned if JWT provided is not registered in the app
Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
200 OK
{
[
{"id": 5,
"owner": "118077088807767242950",
"name": "My New 'Za",
"size": 21,
"style": "Sicilian",
"topping": "",
"self": "https://final-493-wfl.wl.r.appspot.com/pizzas/5"}
],
{"next_link": "n/a"},
{"total_count": 1}
}

401 UNAUTHORIZED
{
"Error: must provide a valid JWT in header"
}

403 FORBIDDEN
{
"Error: JWT provided is not for a registered user"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

Update a Pizza: PUT /pizzas/:id

Request

Path	PUT /pizzas/:id
Accept	application/json
Authorization	Valid JWT as bearer token
Body req.?	Y
Body Format	JSON

Body Attributes:

Name	Description	Required?
name	String providing a name for your pizza; no uniqueness constraint	Y
size	Integer describing pizza's size (diameter in inches)	Y
style	String describing the style of your pizza (generally refers to crust)	Y

Body Example:

```
{
"name": "My New 'Za but Smaller",
"size": 17,
"style": "Still Sicilian"
}
```

Response

Format: JSON

Outcome	Status	Notes
Success	201	Returns updated DB representation of pizza.
Failure	400	Returned if data provided is invalid (i.e. type error, missing fields)
Failure	401	Returned if no valid JWT is provided in request headers
Failure	403	Returned if JWT provided is not the owner of the specified pizza
Failure	404	Returned if ID provided does not match a pizza in the DB

Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
201 CREATED
{
"id": 5,
"owner": "118077088807767242950",
"name": "My New 'Za but Smaller",
"size": 17,
"style": "Still Sicilian",
"topping": "",
"self": "https://final-493-wfl.wl.r.appspot.com/pizzas/5"
}

400 BAD REQUEST
{
"Error: Invalid data provided"
}

401 UNAUTHORIZED
{
"Error: must provide a valid JWT in header"
}

403 FORBIDDEN
{
"Error: Not authorized to access this pizza"
}

404 NOT FOUND
{
"Error: Pizza ID Not Found"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

Update a Pizza (partial): PATCH /pizzas/:id

Request

Path	PATCH /pizzas/:id
Accept	application/json
Authorization	Valid JWT as bearer token
Body req.?	Y
Body Format	JSON

Body Attributes:

Name	**Description**	**Required?**
name	String providing a name for your pizza; no uniqueness constraint	N
size	Integer describing pizza's size (diameter in inches)	N
style	String describing the style of your pizza (generally refers to crust)	N

Body Example:

```
{
"size": 16,
}
```

Response

Format: JSON

Outcome	**Status**	**Notes**
Success	201	Returns updated DB representation of pizza.
Failure	400	Returned if data provided is invalid (i.e. type error, no data)
Failure	401	Returned if no valid JWT is provided in request headers
Failure	403	Returned if JWT provided is not the owner of the specified pizza
Failure	404	Returned if ID provided does not match a pizza in the DB
Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
201 CREATED
{
"id": 5,
"owner": "118077088807767242950",
"name": "My New 'Za but Smaller",
"size": 16,
"style": "Still Sicilian",
"topping": "",
"self": "https://final-493-wfl.wl.r.appspot.com/pizzas/5"
}

400 BAD REQUEST
{
"Error: Invalid data provided"
}

401 UNAUTHORIZED
{
"Error: must provide a valid JWT in header"
}

403 FORBIDDEN
{
"Error: Not authorized to access this pizza"
}

404 NOT FOUND
{
"Error: Pizza ID Not Found"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

Delete a Pizza: DELETE /pizzas/:id

Request

Path	DELETE /pizzas/:id
Authorization	Valid JWT as bearer token

Response

Format: N/A (success), JSON (error)

Outcome	Status	Notes
Success	204	No content; confirms deletion of pizza
Failure	401	Returned if no valid JWT is provided in request headers
Failure	403	Returned if JWT provided is not the owner of the specified pizza
Failure	404	Returned if ID provided does not match a pizza in the DB

Body Examples:

```
204 NO CONTENT

401 UNAUTHORIZED
{
"Error: must provide a valid JWT in header"
}

403 FORBIDDEN
{
"Error: Not authorized to access this pizza"
}

404 NOT FOUND
{
"Error: Pizza ID Not Found"
}
```

Create a Topping: POST /toppings

Request

Path	POST /toppings
Accept	application/json
Body req.?	Y
Body Format	JSON

Body Attributes:

Name	**Description**	**Required?**
name	String providing name for topping; no uniqueness constraint	Y
tastiness	Integer representing how tasty topping is	Y
spiciness	Integer representing how spicy topping is	Y
traditional	Boolean representing whether topping is traditional for pizza	Y

Body Example:

```
{
"name": "Pickled jalapenos",
"tastiness": 9
"spiciness": 8
"traditional": false
}
```

Response

Format: JSON

Outcome	**Status**	**Notes**
Success	201	Creates and returns DB representation, including autogenerated ID.
Failure	400	Returned if data provided is invalid (i.e. type error, missing fields)
Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
201 CREATED
{
"id": 7,
"name": "Pickled jalapenos",
"tastiness": 9,
"spiciness": 8,
"traditional": false,
"self": "https://final-493-wfl.wl.r.appspot.com/toppings/7"
}

400 BAD REQUEST
{
"Error: Invalid data provided"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

View all toppings: GET /toppings

Request

Path	GET /toppings
Accept	application/json

Response

Format: JSON

Outcome	**Status**	**Notes**
Success	200	Returns list of all toppings in DB. Paginated if necessary; also contains total count of toppings
Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
200 OK
{
[
{"id": 7,
"name": "Pickled jalapenos",
"tastiness": 9,
"spiciness": 8,
"traditional": false,
"self": "https://final-493-wfl.wl.r.appspot.com/toppings/7"},
],
{"next_link": "n/a"},
{"total_count": 1}
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

View a topping: GET /toppings/:id

Request

Path	GET /toppings/:id
Accept	application/json

Response

Format: JSON

Outcome	Status	Notes
Success	200	Returns full DB representation of topping with provided ID
Failure	404	Returned if ID provided does not match a topping in the DB
Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
200 OK
{
"id": 7,
"name": "Pickled jalapenos",
"tastiness": 9,
"spiciness": 8,
"traditional": false,
"self": "https://final-493-wfl.wl.r.appspot.com/toppings/7"
}

404 NOT FOUND
{
"Error: Topping ID Not Found"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

Update a topping: PUT /toppings/:id

Request

Path	PUT /toppings/:id
Accept	application/json
Body req.?	Y
Body Format	JSON

Body Attributes:

Name	Description	Required?
name	String providing name for topping; no uniqueness constraint	Y
tastiness	Integer representing how tasty topping is	Y
spiciness	Integer representing how spicy topping is	Y
traditional	Boolean representing whether topping is traditional for pizza	Y

Body Example:

```
{
"name": "Pickled jalapenos",
"tastiness": 10
"spiciness": 8
"traditional": false
}
```

Response

Format: JSON

Outcome	Status	Notes
Success	201	Creates and returns DB representation, including autogenerated ID.
Failure	400	Returned if data provided is invalid (i.e. type error, missing fields)
Failure	404	Returned if ID provided does not match a topping in the DB

Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
201 CREATED
{
"id": 7,
"name": "Pickled jalapenos",
"tastiness": 10,
"spiciness": 8,
"traditional": false,
"self": "https://final-493-wfl.wl.r.appspot.com/toppings/7"
}

400 BAD REQUEST
{
"Error: Invalid data provided"
}

404 NOT FOUND
{
"Error: Topping ID Not Found"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

Update a topping (partial): PATCH /toppings/:id

Request

Path	PATCH /toppings/:id
Accept	application/json
Body req.?	Y
Body Format	JSON

Body Attributes:

Name	Description	Required?
name	String providing name for topping; no uniqueness constraint	N
tastiness	Integer representing how tasty topping is	N
spiciness	Integer representing how spicy topping is	N
traditional	Boolean representing whether topping is traditional for pizza	N

Body Example:

```
{
"name": "Pickled jalapenos (SO GOOD)"
}
```

Response

Format: JSON

Outcome	Status	Notes
Success	201	Creates and returns DB representation, including autogenerated ID.
Failure	400	Returned if data provided is invalid (i.e. type error, no content)
Failure	404	Returned if ID provided does not match a topping in the DB
Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
201 CREATED
{
"id": 7,
"name": "Pickled jalapenos (SO GOOD)",
"tastiness": 10,
"spiciness": 8,
"traditional": false,
"self": "https://final-493-wfl.wl.r.appspot.com/toppings/7"
}

400 BAD REQUEST
{
"Error: Invalid data provided"
}

404 NOT FOUND
{
"Error: Topping ID Not Found"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

Delete a topping: DELETE /toppings/:id

Request

Path	DELETE /toppings/:id

Response

Format: N/A (success), JSON (error)

Outcome	Status	Notes
Success	204	No content; confirms deletion of topping
Failure	404	Returned if ID provided does not match a topping in the DB

Body Examples:

```
204 NO CONTENT

404 NOT FOUND
{
"Error: Topping ID Not Found"
}
```

Add topping to pizza: PUT /pizzas/:pid/topping/:tid

Request

Path	PUT /pizzas/:pid/topping/:tid
Accept	application/json
Authorization	Valid JWT as bearer token
Body req.?	N

Response

Format: JSON

Outcome	Status	Notes
Success	201	Returns updated DB representation of pizza, with topping added
Failure	401	Returned if no valid JWT is provided in request headers
Failure	403	Returned if JWT provided is not the owner of the specified pizza, or specified pizza already has a topping (limit 1)
Failure	404	Returned if IDs provided do not match pizza/topping in the DB
Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
201 CREATED
{
"id": 5,
"owner": "118077088807767242950",
"name": "My New 'Za but Smaller",
"size": 17,
"style": "Still Sicilian",
"topping": "Pickled jalapenos (SO GOOD)",
"self": "https://final-493-wfl.wl.r.appspot.com/pizzas/5"
}

401 UNAUTHORIZED
{
"Error: must provide a valid JWT in header"
}
```

```
403 FORBIDDEN
{
"Error: Not authorized to access this pizza"
"Error: This pizza already has a topping. (Sorry, one per pie!)"
}

404 NOT FOUND
{
"Error: Pizza or Topping ID Not Found"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```

Remove topping from pizza: DELETE /pizzas/:pid/topping/:tid

Request

Path	DELETE /pizzas/:pid/topping/:tid
Accept	application/json
Authorization	Valid JWT as bearer token

Response

Format: JSON

Outcome	Status	Notes
Success	201	Returns updated DB representation of pizza, with topping removed
Failure	400	Returned if topping specified is not present on pizza specified
Failure	401	Returned if no valid JWT is provided in request headers
Failure	403	Returned if JWT provided is not the owner of the specified pizza
Failure	404	Returned if IDs provided do not match pizza/topping in the DB, or topping specified is not present on the pizza currently.
Failure	406	Returned if "Accept: application/json" not in request headers

Body Examples:

```
201 CREATED
{
"id": 5,
"owner": "118077088807767242950",
"name": "My New 'Za but Smaller",
"size": 17,
"style": "Still Sicilian",
"topping": "",
"self": "https://final-493-wfl.wl.r.appspot.com/pizzas/5"
}

401 UNAUTHORIZED
{
"Error: must provide a valid JWT in header"
}
```

```
403 FORBIDDEN
{
"Error: Not authorized to access this pizza"
}

404 NOT FOUND
{
"Error: Pizza or Topping ID Not Found"
"Error: Topping not present on this pizza"
}

406 NOT ACCEPTABLE
{
"Error: Accept header must be configured for application/json"
}
```