Assignment 3: API Spec

CS 493: Cloud Application Development

Oregon State University

Change Log

Version	Change	Date
1.0	Initial version.	Oct 16, 2024

Note: I have highlighted in yellow the major differences between this specification and the API Specification for Assignment 2.

Create a Business

Allows you to create a new business.

```
POST /businesses
```

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
owner_id	The ID of the owner. A number	Yes
name	The name of the business. Up to 50 characters.	Yes
street_address	Street address of the business. Up to 100 characters.	Yes
city	City. Up to 50 characters.	Yes
state	Two letter code for the state	Yes
zip_code	5 digit zip code.	Yes

Request Body Example

```
{
  "owner_id": 123,
  "name": "Mandola's",
  "street_address": "4900 N Lamar Blvd",
  "city": "Austin",
  "state": "TX",
  "zip_code": 78751
}
```

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 6 required attributes, the business must not be created, and 400 status code must be returned. You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid.

		You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).

Response Examples

- The REST API generates the value of id which must be a unique value for businesses.
- Response must include a property **self** whose value is the complete URL for this business.

Success

```
Status: 201 Created

{
  "id": 156789,
  "owner_id": 123,
  "name": "Mandola's",
  "street_address": "4900 N Lamar Blvd",
  "city": "Austin",
  "state": "TX",
  "zip_code": 78751,
    "self": "http://104.198.173.141:8000/businesses/156789"
}
```

Failure

```
Status: 400 Bad Request

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

Get a Business

Allows you to get an existing business

```
GET /businesses/:business_id
```

Request

Path Parameters

Name	Description
business_id	ID of the business

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No business with this business_id exists

Response Examples

- Response must include a property **self** whose value is the complete URL for this business

Success

```
Status: 200 OK

{
  "id": 156789,
  "owner_id": 123,
  "name": "Mandola's",
  "street_address": "4900 N Lamar Blvd",
  "city": "Austin",
  "state": "TX",
  "zip_code": 78751,
   "self": "http://104.198.173.141:8000/businesses/156789"
}
```

Failure

```
Status: 404 Not Found

{
"Error": "No business with this business_id exists"
}
```

List all Businesses

List all the businesses.

```
GET /businesses
GET /businesses?offset=3&limit=3
…
```

Request

Path Parameters

None

Query Parameters

- 2 optional query parameters. Note that the order of query parameters doesn't matter.
- You can assume the following:
  - A request will either have both parameters or neither.
  - The value of both will be an integer. You don't need to validate the value.

Name	Description
offset	Offset.
limit	Limit. Value will be 3

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

- Each business in the response must include a property **self** whose value is the complete URL for that business.
- This endpoint implements pagination with a page size of 3.
- The property "entries" is an array with the businesses for this page.
- The property "next" is a URL with query parameters offset and limit set to correct values.
  - Optional: If a page has fewer than 3 entries then it is certainly the last page and it should not have the "next" link (this is optional to implement and we are not going to deduct points for it)

Success

```
Status: 200 OK
{
"entries": [
   {
    "city": "Austin",
    "id": 5631671361601536,
    "name": "Mandola's",
    "owner_id": 1234123,
    "state": "TX",
    "street_address": "4900 N Lamar Blvd",
    "zip_code": 78751,
    "self": "http://104.198.173.141:8000/businesses/5631671361601536"
  },
  {
    "city": "Austin",
    "id": 5632499082330112,
    "name": "Torchy Tacos",
    "owner_id": 412349834,
    "state": "TX",
    "street_address": "5024 Burnet Road",
    "zip_code": 78754,
   "self": "http://104.198.173.141:8000/businesses/5632499082330112"
  },
  {
    "city": "Austin",
    "id": 5631671361601488,
    "name": "Mandola's",
    "owner_id": 1234123,
    "state": "TX",
    "street_address": "4900 N Lamar Blvd",
    "zip_code": 78751,
    "self": "http://104.198.173.141:8000/businesses/5631671361601488"
  },
],
 "next": "http://104.198.173.141:8000/businesses?offset=3&limit=3"
}
```

Edit a Business

Allows you to edit a business. Replacement semantics.

```
PUT /businesses/:business_id
```

Request

Path Parameters

Name	Description
business_id	ID of the business

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
owner_id	The ID of the owner. A number	Yes
name	The name of the business.	Yes
street_address	Street address of the business.	Yes
city	City	Yes
state	Two letter code for the state	Yes
zip_code	5 digit zip code.	Yes

Request Body Example

```
{
  "owner_id": 123,
  "name": "Mandola's Italian",
  "street_address": "4900 N Lamar Blvd",
  "city": "Austin",
  "state": "TX",
  "zip_code": 78751
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing any of the 6 required attributes, the business must not be updated, and 400 status code must be returned. You don’t need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute’s value is valid.

		You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute other than the ones that are listed).
Failure	404 Not Found	No business with this business_id exists

Response Examples

- Response must include a property **self** whose value is the complete URL for this business.

Success

```
Status: 200 OK

{
 "id": 156789,
 "owner_id": 123,
 "name": "Mandola's Italian",
 "street_address": "4900 N Lamar Blvd",
 "city": "Austin",
 "state": "TX",
 "zip_code": 78751,
  "self": "http://104.198.173.141:8000/businesses/156789"
}
```

Failure

```
Status: 400 Bad Request

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

```
Status: 404 Not Found

{
"Error": "No business with this business_id exists"
}
```

Delete a Business

Allows you to delete a business. Note that if there are any reviews for a business, deleting the business also deletes those reviews.

```
DELETE /businesses/:business_id
```

Request

Path Parameters

Name	Description
business_id	ID of the business

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 business with this business_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

{
"Error": "No business with this business_id exists"
}
```

List all Businesses for an Owner

List all the businesses for an owner.

```
GET /owners/:owner_id/businesses
```

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

- Each business in the response must include a property **self** whose value is the complete URL for that business.

Success

```
Status: 200 OK

[
   {
      "city": "Austin",
      "id": 5631671361601536,
      "name": "Torchy's",
      "owner_id": 412349834,
      "state": "TX",
      "street_address": "5100 Mueller Drive",
      "zip_code": 78761,
      "self": "http://104.198.173.141:8000/businesses/5631671361601536"
   },
   {
      "city": "Austin",
      "id": 5632499082330112,
      "name": "Torchy Tacos",
      "owner_id": 412349834,
      "state": "TX",
      "street_address": "5024 Burnet Road",
      "zip_code": 78754,
      "self": "http://104.198.173.141:8000/businesses/5632499082330112"
```

```
    }
]
```

Create a Review

Allows you to create a new review.

```
POST /reviews
```

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
user_id	ID of the user. A number	Yes
business_id	ID of the business	Yes
stars	Number of stars, 0 to 5 (inclusive)	Yes
review_text	String. Up to 1000 characters	No

Request Body Example

```
{
  "user_id": 12134,
  "business_id": 156789,
  "stars": 4,
  "review_text":   "Mandola’s has great pasta"
}
```

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, the business must not be created, and 400 status code must be returned. You don’t need to validate the values of the attributes user_id, starts and review_text can assume that if the request contains any of the these attributes, then that attribute’s value is valid. However, you cannot assume that a business with the provided value of business_id exists. If no such business exists, 404 status code must be returned.

		You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).
Failure	404 Not found	No business exists with the provided business_id
Failure	409 Conflict	A previous review already exists by this user for this business.

Response Examples

- The REST API generates the value of id which must be a unique value for reviews.
- Response must include:
  - A property **self** whose value is the complete URL for this review.
  - A property **business** whose value is the complete URL for the business corresponding to this review.

Success

```
Status: 201 Created

{
  "id": 19884,
   "user_id": 12134,
   "business": "http://104.198.173.141:8000/businesses/156789"
   "stars": 4,
   "review_text":   "Mandola's has great pasta",
    "self": "http://104.198.173.141:8000/reviews/19884"
}
```

Failure

```
Status: 400 Bad Request

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

```
Status: 404 Not found

{
"Error":  "No business with this business_id exists"
}
```

```
Status: 409 Conflict

{
"Error":  "You have already submitted a review for this business. You can update
your previous review, or delete it and submit a new review"
}
```

Get a Review

Allows you to get an existing review.

```
GET /reviews/:review_id
```

Request

Path Parameters

Name	Description
review_id	ID of the review

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No review with this review_id exists

Response Examples

- Response must include:
  - A property **self** whose value is the complete URL for this review.
  - A property **business** whose value is the complete URL for the business corresponding to this review.

Success

```
Status: 200 OK

{
  "id": 19884,
   "user_id": 12134,
   "business": "http://104.198.173.141:8000/businesses/156789"
   "stars": 4,
   "review_text":   "Mandola’s has great pasta",
    "self": "http://104.198.173.141:8000/reviews/19884"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No review with this review_id exists"
}
```

Edit a Review

Allows you to edit a review. Partial update semantics. Only update the fields that are in the request.

```
PUT /reviews/:review_id
```

Request

Path Parameters

Name	Description
review_id	ID of the review

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
stars	Number of stars, 0 to 5 (inclusive)	Yes
review_text	String	No.

Request Body Example

```
{
  "stars": 4,
  "review_text":   "Mandola’s has great pasta. Mista salad is great too!"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing the attribute stars, the review must not be updated, and 400 status code must be returned. You don’t need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute’s value is valid. You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute other than the ones that are listed).
Failure	404 Not Found	No review with review_id exists

Response Examples

- Response must include:
  - A property **self** whose value is the complete URL for this review.

- A property **business** whose value is the complete URL for the business corresponding to this review.

Success

```
Status: 200 OK

{
 "id": 19884,
  "user_id": 12134,
  "business": "http://104.198.173.141:8000/businesses/156789",
  "stars": 4,
  "review_text":   "Mandola’s has great pasta. Mista salad is great too!",
   "self": "http://104.198.173.141:8000/reviews/19884"
}
```

Failure

```
Status: 400 Bad Request

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

```
Status: 404 Not Found

{
"Error": "No review with this review_id exists"
}
```

Delete a Review

Allows you to delete a review.

```
DELETE /reviews/:review_id
```

Request

Path Parameters

Name	Description
review_id	ID of the review

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 review with this review_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

{
"Error": "No review with this review_id exists"
}
```

List all Reviews for a User

List all the reviews.

```
GET /users/:user_id/reviews
```

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

- Each review in the response must include:
  - A property **self** whose value is the complete URL for this review.
  - A property **business** whose value is the complete URL for the business corresponding to this review.

Success

```
Status: 200 OK

[
   {
          "id": 19884,
           "user_id": 12134,
          "business": "http://104.198.173.141:8000/businesses/156789",
           "stars": 4,
           "review_text": "Mandola's has great pasta.",
          "self": "http://104.198.173.141:8000/reviews/19884"
  }
  {
          "id": 69884,
           "user_id": 21342,
            "business": "http://104.198.173.141:8000/businesses/156790",
           "stars": 4,
           "review_text": "Shrimp taco is the best!",
           "self": "http://104.198.173.141:8000/reviews/69884"
 }
]
```