

API Spec

Last Update: 6/21/2022

Change log	1
Data Model.....	2
Create a DVD.....	3
View a DVD	4
View all DVDs.....	5
Update a DVD with PUT.....	6
Update a DVD with PATCH.	8
PUT all DVDs (not allowed)	10
DELETE all DVDs (not allowed).....	10
Create a Customer.....	11
View a Customer.....	13
View all Customers.....	14
PUT all Customers (not allowed)	15
Rent a DVD to a Customer	16
Return a DVD from a Customer	17
DELETE all Customers (not allowed).....	18
Update a Customer with PUT.	18
Update a Customer with PATCH.	19
Delete a Customer.....	21
View all Store Managers (Users).....	22

Change log

Version	Change	Date
1.0	Initial version.	June 5, 2022
1.1	Minor formatting updates.	June 21, 2022

Data Model

The app stores three kinds of entities in Datastore: DVDs, Customers and Store Managers (users).

DVD – protected

Property	Data Type	Notes
id	Integer	The id of the DVD. Datastore automatically generates it.
name	String	Name of the DVD.
genre	String	Type of the DVD. E.g., Comedy, Drama, etc.
length	Integer	The length of the DVD in minutes.
renter	JSON	The customer that is renting the DVD.
store_manager	JWT	The store manager/user that owns the DVD.

Customer

Property	Data Type	Notes
id	Integer	The id of the Customer. Datastore automatically generates it.
name	String	The name of the Customer
member	Boolean	Identifies where Customer is a member, in which case they could be eligible for discounts.
rentals	List	List of DVDs currently rented to Customer.
birth_month	String	Month the customer was born.

Store Manager

Property	Data Type	Notes
user_id	Integer	The id of the Store Manager, sub from JWT
name	String	The name of the Store Manager (from Google OAuth)

Create a DVD

Allows Store Manager (user) to create a new DVD.

POST /dvds

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the DVD.	Yes
genre	Type of the DVD. E.g., Comedy, Drama, etc.	Yes
length	The length of the DVD in minutes.	Yes

Request Body Example

```
{
  "name": "Zoolander",
  "genre": "Comedy",
  "length": 90,
}
```

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 required attributes, the DVD will not be created, and 400 status code is returned.
Failure	401 Unauthorized	If the JWT is invalid, DVD will not be created.
Failure	406 Not Acceptable	If the client sends a request with Accept header set to a MIME type that is not JSON.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- Self is generated dynamically
- Renter defaults to null and will be updated when a Customer rents the DVD

Success

```
Status: 201 Created

{
  "id": 123,
```

```
"name": "Zoolander",
"genre": "Comedy",
"length": 90,
"store_manager": "11655246556562509061",
"self" : "https://appspot.com/dvds/1234"
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

```
{
  "Error": "You must be a valid user to create a DVD. "
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Response type must be JSON"
}
```

View a DVD

Allows you to view an existing DVD

GET /dvds/:dvd_id

Request

Path Parameters

Name	Description
DVD_id	ID of the DVD

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No DVD with this DVD_id exists
Failure	401 Unauthorized	If the JWT is invalid, DVD cannot be viewed.
Failure	406 Not Acceptable	If the client sends a request with Accept header set to a MIME type that is not JSON.

Response Examples

Success

Status: 200 OK
<pre>"id": 123, "name": "Zoolander", "genre": "Comedy", "length": 90, "renter": null, "store_manager": "11655246556562509061", "self" : "https://appspot.com/Dvds/1234" }</pre>

Failure

Status: 401 Unauthorized
<pre>{ "Error": "You must be a valid store manager to view this DVD. " }</pre>

Status: 406 Not Acceptable
<pre>{ "Error": "Response type must be JSON" }</pre>

View all DVDs

List all the DVDs. Uses pagination to return 5 Dvds per page and includes a “next” link to get the next page of results. If valid JWT provided, will only display Dvds owned by that user.

GET /dvds

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 the JWT is invalid, Dvds cannot be viewed.
Failure	406 Not Acceptable	If the client sends a request with Accept header set to a

		MIME type that is not JSON.
--	--	-----------------------------

Response Examples

Success

```
{
  "dvds": [
    {
      "store_manager": "116552465565625090651",
      "renter": null,
      "length": 108,
      "genre": "Musical",
      "name": "Sing",
      "id": 4787246431469568
    },
    {
      "store_manager": "116552465565625090651",
      "length": 109,
      "renter": null,
      "genre": "Horror",
      "name": "Psycho",
      "id": 4802063531769856
    },
    {
      "name": "Monsters Inc",
      "genre": "Cartoon",
      "store_manager": "116552465565625090651",
      "length": 96,
      "renter": null,
      "id": 5394119539032064
    },
    {
      "length": 95,
      "renter": null,
      "store_manager": "116552465565625090651",
      "name": "A Bug's Life",
      "genre": "Cartoon",
      "id": 5913146338312192
    },
    {
      "genre": "Romantic Comedy",
      "store_manager": "116552465565625090651",
      "length": 119,
      "renter": null,
      "name": "Pretty Woman",
      "id": 6199595021369344
    }
  ],
  "next": "http://localhost:8080/dvds?limit=5&offset=5"
}
```

Failure

Status: 401 Unauthorized

```
{
  "Error": "You must be a valid store_manager to view Dvds. "
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Response type must be JSON"
}
```

Update a DVD with PUT.

A DVD is updated.

PUT Dvds/:DVD_id/

Request

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the DVD.	Yes

genre	Type of the DVD. E.g., Comedy, Drama, etc.	Yes
length	The length of the DVD in minutes.	Yes
renter	Null or JSON of customer that is renting the DVD	Yes

Request Body Example

```
{
  "name": "Zoolander Updated",
  "genre": "Comedy",
  "length": 91,
  "renter": null,
}
```

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If JWT is not the owner of DVD, the DVD cannot be updated.
Failure	404 Not Found	No DVD with this DVD_id exists
Failure	406 Not Acceptable	If the client sends a request with Accept header set to a MIME type that is not JSON.

Response Examples

Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Zoolander Updated",
  "genre": "Comedy",
  "length": 91,
  "renter": null,
  "store_manager": "116552465565625090651",
  "self" : "https://appspot.com/Dvds/1234"
}
```

Failure

Status: 401 Unauthorized

```
{
  "Error": "You must be a valid store manager to update this DVD. "
}
```

Status: 404 Not Found

```
{
  "Error": "No DVD with this DVD_id exists"
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Response type must be JSON"  
}
```

Update a DVD with PATCH.

A DVD is updated.

PATCH Dvds/:DVD_id/

Request

Request Body

At least one attribute must be included to patch.

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	Name of the DVD.	No
genre	Type of the DVD. E.g., Comedy, Drama, etc.	No
length	The length of the DVD in minutes.	No
renter	Null or JSON of customer that is renting the DVD	No

Request Body Example

```
{  
  "name": "Zoolander Extended",  
  "length": 95  
}
```

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If there is not at least one attribute in the request to PATCH, error 400 will be returned.
Failure	401 Unauthorized	If JWT is not the owner of DVD, the DVD cannot be updated.
Failure	404 Not Found	No DVD with this DVD_id exists
Failure	406 Not Acceptable	If the client sends a request with Accept header set to a MIME type that is not JSON.

Response Examples

Success

Status: 200 OK

```
"id": 123,
```



```
{
  "name": "Zoolander Extended",
  "genre": "Comedy",
  "length": 95,
  "renter": null,
  "store_manager": "116552465565625090651",
  "self" : "https://appspot.com/Dvds/1234"
}
```

Failure

Status: 401 Unauthorized

```
{
  "Error": "You must be a valid user to update this DVD. "
}
```

Status: 404 Not Found

```
{
  "Error": "No DVD with this DVD_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Response type must be JSON"
}
```

Delete a DVD

Allows you to delete a DVD. If the DVD is rented by a Customer, the Customer will have the DVD returned (and removed from their list of rentals).

DELETE /dvds/:dvd_id

Request

Path Parameters

Name	Description
DVD_id	ID of the DVD

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 not the owner of DVD, the DVD cannot be deleted.
Failure	404 Not Found	No DVD with this DVD_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{  
  "Error": "You must be a valid store manager to delete this DVD. "  
}
```

Status: 404 Not Found

```
{  
  "Error": "No DVD with this DVD_id exists"  
}
```

PUT all DVDs (not allowed)

PUT /dvds

Request

Path Parameters

None

Response Body Format

Failure: JSON

Failure

Status: 405 Method Not Allowed

```
{  
  "Error": "Cannot apply PUT to all Dvds"  
}
```

DELETE all DVDs (not allowed)

DELETE /dvds

Request

Path Parameters

None

Response Body Format

Failure: JSON

Failure

Status: 405 Method Not Allowed

```
{
  "Error": "Cannot apply DELETE to all Dvds"
}
```

Create a Customer

Allows you to create a new Customer.

POST /Customers

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the Customer	Yes
member	Identifies where Customer is a member, in which case they could be eligible for discounts.	Yes
rentals	List of Dvds currently rented to Customer.	Yes
birth_month	The month the Customer was born.	yes

Request Body Example

```
{
  "name": "Jess Member",
  "member": 1,
  "birth_month": "March"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing the name, member, rentals or birth_month attribute, the Customer will not be created, and 400 status code will be returned.
Failure	406 Not Acceptable	If the client sends a request with Accept header set to a MIME type that is not JSON.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created
- Rentals defaults to [] until customer rents at least 1 DVD
- Self is generated dynamically

Success

```
Status: 201 Created
{
  "id" : 50134,
  "name": "Jess Member",
  "member": 1,
  "rentals": [],
  "birth_month": "March",
  "self" : "http://localhost:8080/customers/5187754950393856"
}
```

Failure

```
Status: 400 Bad Request

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

Status: 406 Not Acceptable

```
{
  "Error": "Response type must be JSON"
}
```

View a Customer

Allows you to view an existing Customer.

GET /Customers/:Customer_id

Request

Path Parameters

Name	Description
Customer_id	ID of the Customer

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No Customer with this Customer_id exists
Failure	406 Not Acceptable	If the client sends a request with Accept header set to a MIME type that is not JSON.

Response Examples

Success

Status: 200 OK

```
{
  "id": 50134,
  "name": "Jess Member",
  "member": 1,
  "rentals": [],
  "birth_month": "April"
}
```

Failure

Status: 404 Not Found

```
{
  "Error": "No Customer with this Customer_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Response type must be JSON"
}
```

View all Customers

List all the Customers. Uses pagination to return 5 Customers per page and includes a “next” link to get the next page of results.

GET /Customers

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If the client sends a request with Accept header set to a MIME type that is not JSON.

Response Examples

Success

```
Status: 200 OK
{"Customers":
{
  "id" : 50134,
  "name": "Jess Member",
  "member": 1,
  "rentals": []
  "birth_month": "March"
},
{
  "id" : 50135,
  "name": "Jess Smith",
  "member": 0,
  "rentals": []
  "birth_month": "May"
}
,
{
  "id" : 50138,
  "name": "Mickey Member",
```

```
"member": 1,
"rentals": []
"birth_month": "August"
}
,
{
"id" : 50122,
"name": "Hamilton Member",
"member": 1,
"rentals": []
"birth_month": "November"
}
,
{
"id" : 501384,
"name": "Don Customer",
"member": 0,
"rentals": []
"birth_month": "December"
}

"next": "https://duttonj-dvds.uw.r.appspot.com/Customers?limit=5&offset=5"}]
```

Failure

Status: 406 Not Acceptable

```
{
"Error": "Response type must be JSON"
}
```

PUT all Customers (not allowed)

PUT /Customers

Request

Path Parameters

None

Response Body Format

Failure: JSON

Failure

Status: 405 Method Not Allowed

```
{
"Error": "Cannot apply PUT to all Customers"
}
```

Rent a DVD to a Customer

A Customer rents a DVD.

PUT / Customers/:Customer_id/ Dvds/:DVD_id

Request

Path Parameters

Name	Description
Customer_id	ID of the Customer
DVD_id	ID of the DVD

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 DVD exists with this DVD_id, a Customer exists with this Customer_id and the DVD is rented to the Customer.
Failure	401 Unauthorized	If JWT is not the owner of DVD, the DVD cannot be returned.
Failure	403 Forbidden	DVD is already rented by a customer.
Failure	404 Not Found	The specified DVD doesn't exist, the customer doesn't exist.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
  "Error": "You must be a valid store manager to rent this DVD. "
}
```

Status: 403 Forbidden

```
{
  "Error": "This DVD is not available to rent"
}
```

Status: 404 Not Found

```
{
  "Error": "This DVD and/or this customer does not exist"}

```


Return a DVD from a Customer

A Customer returns a DVD.

DELETE / Customers/:Customer_id/ Dvds/:DVD_id

Request

Path Parameters

Name	Description
Customer_id	ID of the Customer
DVD_id	ID of the DVD

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 DVD exists with this DVD_id, a Customer exists with this Customer_id and the DVD is rented to the Customer.
Failure	401 Unauthorized	If JWT is not the owner of DVD, the DVD cannot be returned.
Failure	404 Not Found	The specified DVD doesn't exist, the customer doesn't exist, or the DVD is not rented by the Customer.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
  "Error": "You must be a valid store manager to return this DVD. "
}
```

Status: 404 Not Found

```
{
  "Error": "No DVD with this DVD_id is rented by the Customer with this Customer_id"
}
```

DELETE all Customers (not allowed)

DELETE /Customers

Request

Path Parameters

None

Response Body Format

Failure: JSON

Failure

Status: 405 Method Not Allowed

```
{
  "Error": "Cannot apply DELETE to all Customers"
}
```

Update a Customer with PUT.

A Customer is updated.

PUT Customers/:Customer_id/

Request

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the Customer	Yes
member	Identifies where Customer is a member, in which case they could be eligible for discounts.	Yes
rentals	List of Dvds currently rented to Customer.	Yes
birth_month	The month the Customer was born.	Yes

Request Body Example

```
{
  "name": "Don Nowmember",
  "member": 1,
  "rentals": [],
  "birth_month" : "August"
}
```

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	403 Forbidden	Cannot rent or return DVDs at this URL.
Failure	404 Not Found	No Customer with this Customer_id exists
Failure	406 Not Acceptable	If the client sends a request with Accept header set to a MIME type that is not JSON.

Response Examples

Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Don Nowmember",
  "member": 1,
  "rentals": [],
  "birth_month" : "August"
  "self" : "https://appspot.com/customers/1234"
}
```

Failure

Status: 403 Forbidden

```
{
  "Error": "Cannot rent or return DVDs at this URL."
}
```

Status: 404 Not Found

```
{
  "Error": "No Customer with this Customer_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Response type must be JSON"
}
```

Update a Customer with PATCH.

A CUSTOMER is updated.

PATCH Customer s/: Customer_id/

Request

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the Customer	No
member	Identifies where Customer is a member, in which case they could be eligible for discounts.	No
rentals	List of Dvds currently rented to Customer.	No
birth_month	The month the Customer was born.	No

Request Body Example

<pre>{ "name": "Hamilton Smith", "member": 0, }</pre>

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	403 Forbidden	Cannot rent or return DVDs at this URL.
Failure	404 Not Found	No Customer with this Customer_id exists
Failure	406 Not Acceptable	If the client sends a request with Accept header set to a MIME type that is not JSON.

Response Examples

Success

Status: 200 OK
<pre>{ "id": 124, "name": "Hamilton Smith", "member": 0, "rentals": [], "birth_month": "August", "self": "https://appspot.com/customers/124" }</pre>

Failure

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

```
"Error": "Cannot rent or return DVDs at this URL."
}
```

Status: 404 Not Found

```
{
  "Error": "No Customer with this Customer_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Response type must be JSON"
}
```

Delete a Customer

Allows you to delete a Customer. If a Customer has rentals, the rentals will be updated.

DELETE /Customers/:Customer_id

Request

Path Parameters

Name	Description
Customer_id	ID of the Customer

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 Customer with this Customer_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
  "Error": "No Customer with this Customer_id exists"
}
```

View all Store Managers (Users)

List all the Store Managers (users).

GET /Users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If the client sends a request with Accept header set to a MIME type that is not JSON.

Response Examples

Success

```
Status: 200 Success
{"List of store manager IDs": [1657979654654]}
```

Failure

```
Status: 406 Not Acceptable

{
  "Error": "Response type must be JSON"
}
```