

Backend 3.3_CW Exercises

ex01: add descriptive title

1. Add all your car related api requests to the Car Inventory management folder.
2. Click on the request to which you want to add a description & choose "View documentation" from the right panel.
3. At the top of the request details, you'll see the request name or title.
4. Click on the title to edit it.
5. Write a clear and concise title that summarises the purpose of the request(e.g., "Retrieve All Cars").

ex02: write a detailed description

1. Below the title, you'll find a text field or description area.
2. Write a detailed description that provides context for the request.

Example:

Title: Retrieve All Cars

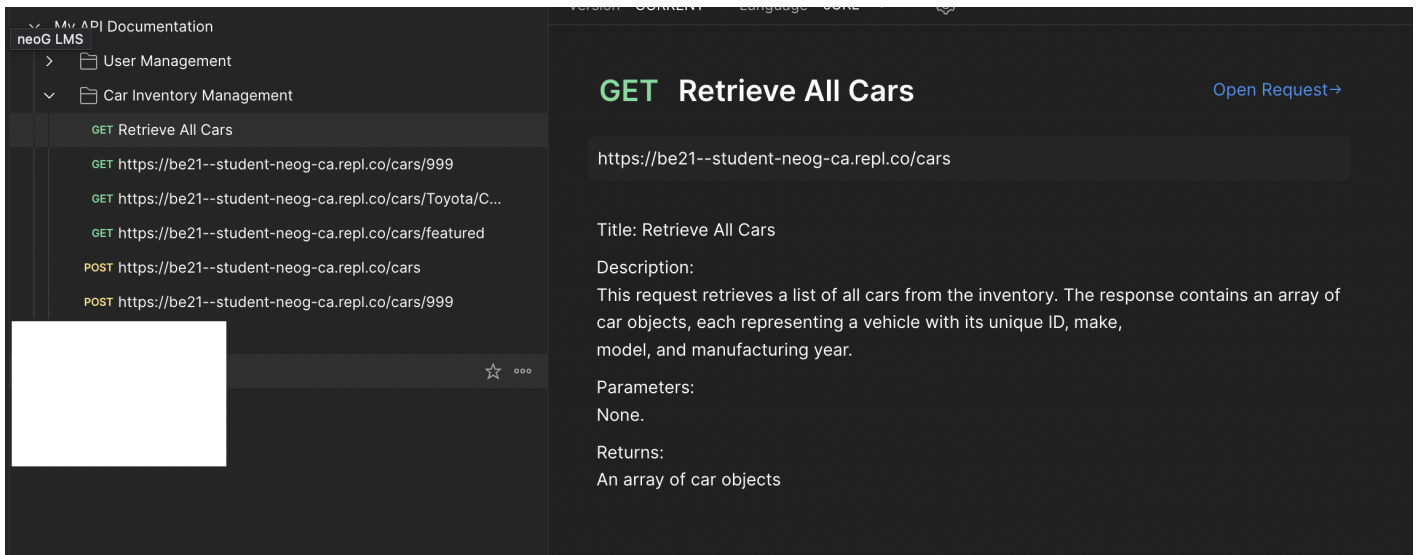
Description: This request retrieves a list of all cars from the inventory. The response contains an array of car objects, each representing a vehicle with its unique ID, make, model, and manufacturing year.

Parameters: None.

Returns: An array of car objects

COPY

Solution:



ex02.1: description having parameters

challenge

Write description for the below request:

Modify GET Route for Cars: Update the `/cars` route to handle query parameters.

```
app.get('/cars', (req, res) => {
  const { make, model } = req.query
  let filteredCars = cars

  if (make) {
    filteredCars = filteredCars.filter(
      (car) => car.make.toLowerCase() === make.toLowerCase(),
    )
  }
  if (model) {
    filteredCars = filteredCars.filter(
      (car) => car.model.toLowerCase() === model.toLowerCase(),
    )
  }

  res.json(filteredCars)
})
```

COPY

solution

****Title:** Filter Cars by Make and Model**

****Description:**** This request allows you to filter the list of cars based on specific criteria. Provide query parameters to filter cars by their make or model. The response will include an array of car objects that match the specified criteria.

****Parameters:**** ``make`` (optional, string): Filter cars by manufacturer (e.g., Toyota, Honda). ``model`` (optional, string): Filter cars by model name (e.g., Camry, Civic).

****Returns:**** Response Body: An array of car objects that match the provided criteria.

COPY

ex02.2: description having parameters w/ header

challenge

Write description for the below request:

Create GET Route with Headers: Define a GET route at /cars/featured that returns featured cars based on a custom header.

```
app.get('/cars/featured', (req, res) => {
  const isFeaturedRequest = req.header('x-featured-request')

  if (isFeaturedRequest === 'true') {
    const featuredCars = cars.filter((car) => car.year >= 2021)
    res.json(featuredCars)
  } else {
    res.status(400).json({ error: 'Invalid request' })
  }
})
```

COPY

solution

****Title:** Retrieve Featured Cars

****Description:**** This request fetches a list of featured cars from the inventory based on a custom header ****`x-featured-request`****. If the header value is set to ****`true`****, the response will include an array of cars manufactured in or after the year **2021**. If the header value is not provided or is set to anything other than ****`true`****, a **400** error response will be returned.

****Parameters:****

- Header: ****`x-featured-request`**** (boolean, optional): Set to ****`true`**** to request featured cars.

****Returns:****

- Response Body: An array of car objects representing featured cars.