**Car Application Documentation**

**Overview**

The Car application is designed to manage car-related operations, including pricing calculations, logging, and API endpoints. Below are the key features and implementation details.

Features

1. In-Memory Database (Entity Framework):

* Implemented an in-memory database using Entity Framework (EF).
* Created base tables and established relations.

1. Logging with SeriLog:

* Integrated SeriLog for comprehensive logging.
* Initial configuration includes console and file logging.
* Consider future extensions (e.g., ElasticSearch).

1. Price Calculation Strategy:

* Utilized the strategy pattern for price calculation.
* Covered by xUnit tests to ensure correctness.

1. Code Organization and Readability:

* Centralized Car-related code in CarRequests.cs.
* Introduced a ValidatorExtension for FluentValidation.
* Structured the web API using a chain pattern.

1. Unit Tests:

* Demonstrated unit tests for price calculation and validator.
* Ensured robust validation and accurate results.

1. XML Documentation:

* Created an XML file (Hedin.ChangeTires.Api.xml) for future reference.
* Useful for generating API documentation.

1. Token-Based Authentication and Swagger Integration:

* Implemented token-based authentication.
* Tested authentication within the Swagger interface.

1. Swagger Documentation:

* Documented API endpoints in Swagger.
* Included example code snippets and request/response details.

1. External APIs:

* Developed two external APIs.
* Concealed implementation details.

1. Error Handling (Try-Catch):

* Demonstrated try-catch blocks for robust error handling.

1. Secure Dirty Reads (Database):

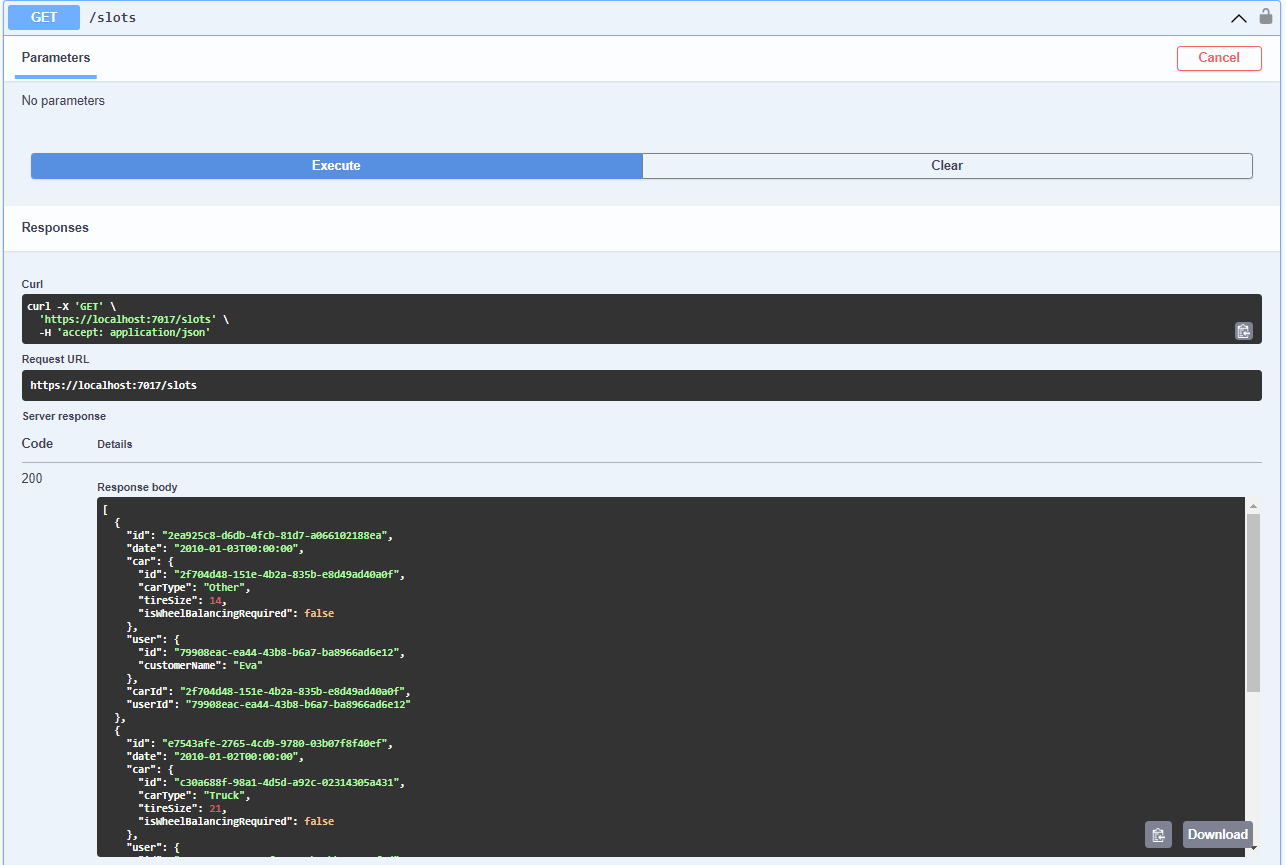
* Note that dirty reads are feasible only in a real database (not in-memory).

1. Project Scope and Focus:

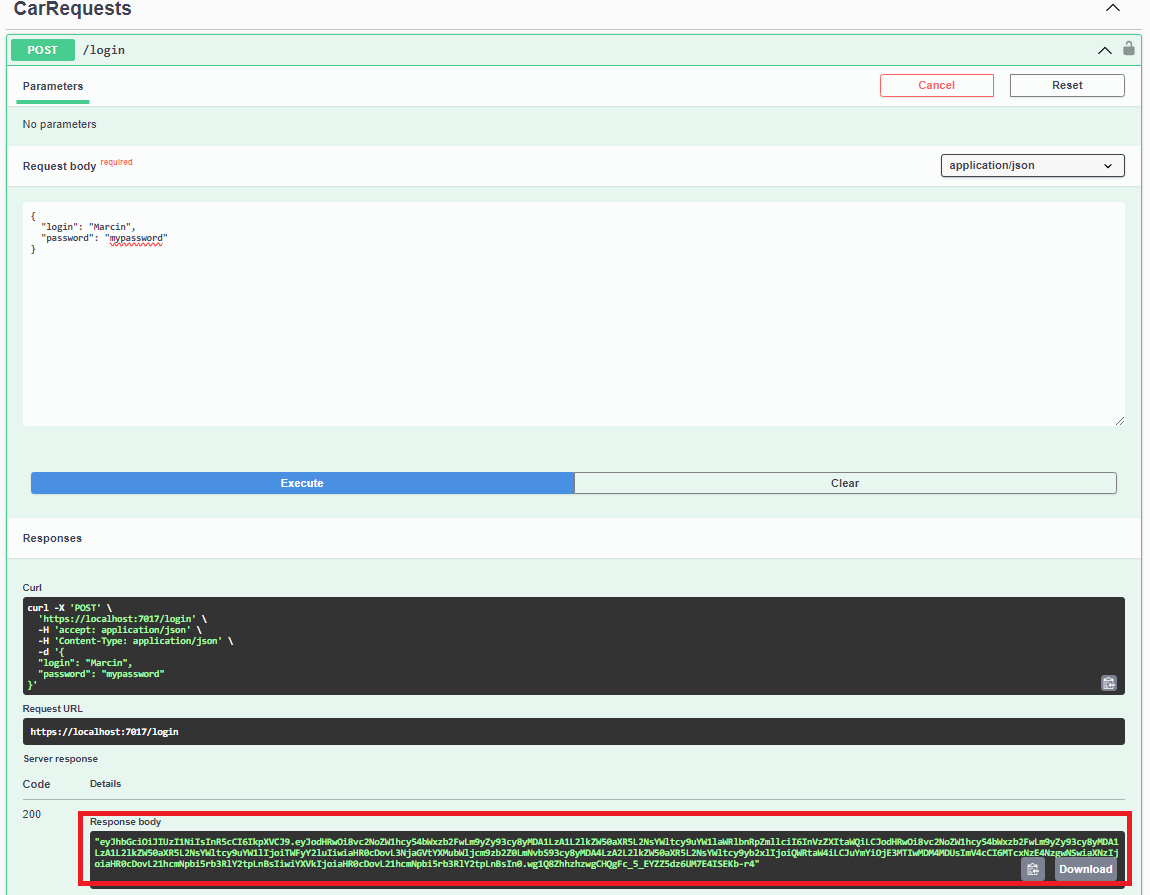
* Acknowledged time limitations.
* Emphasized code approaches over advanced services.
* Prepared for in-depth discussions during interviews.

Example case application:

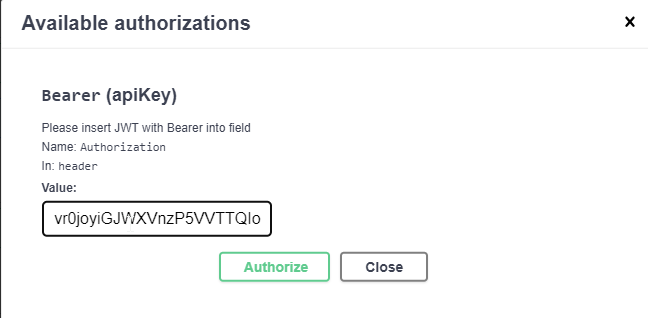
1. Chceck booked slots (one picture of more than a thousand words)



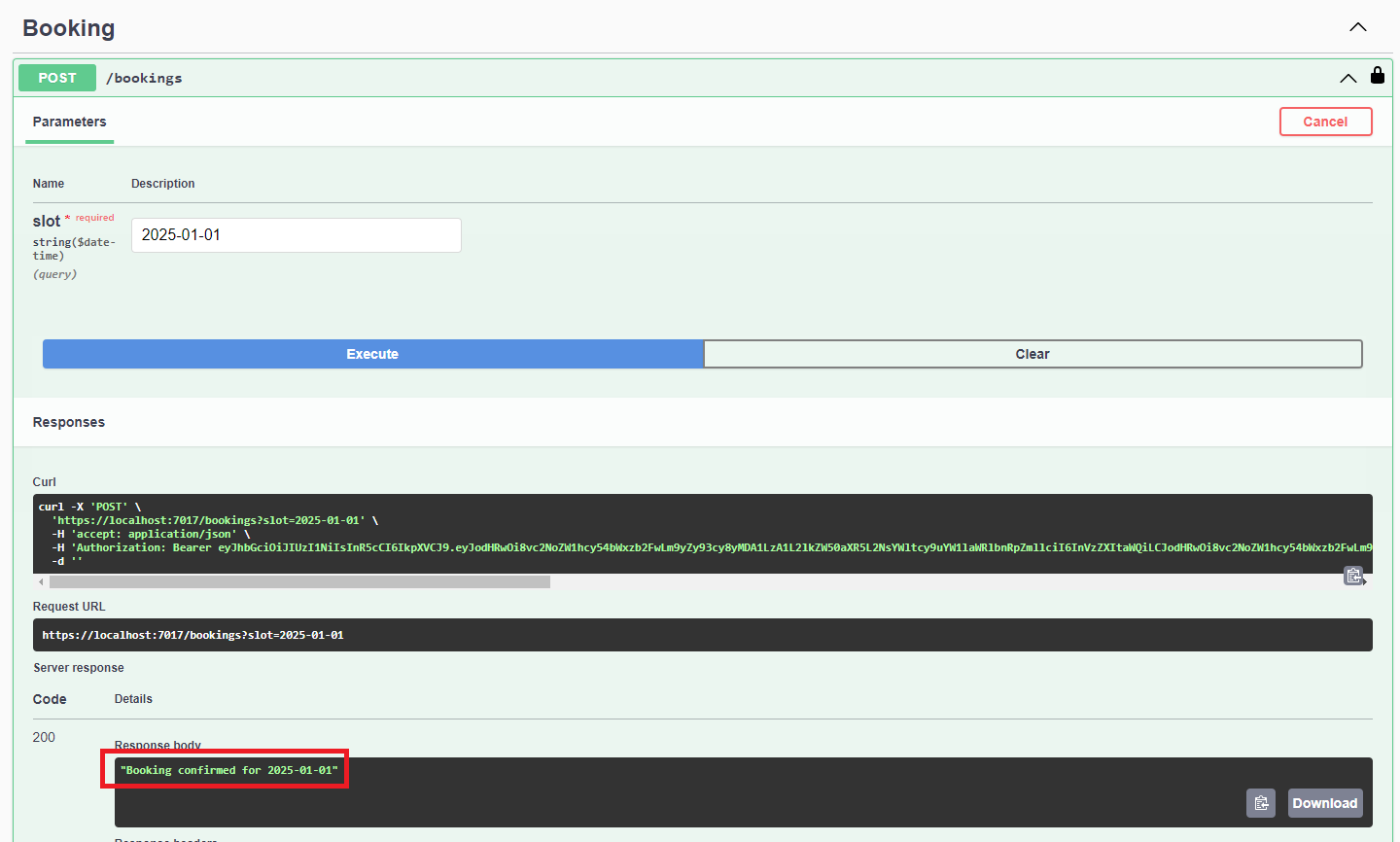
1. Booking slot
   1. Get token (login: Marcin password: mypassword)



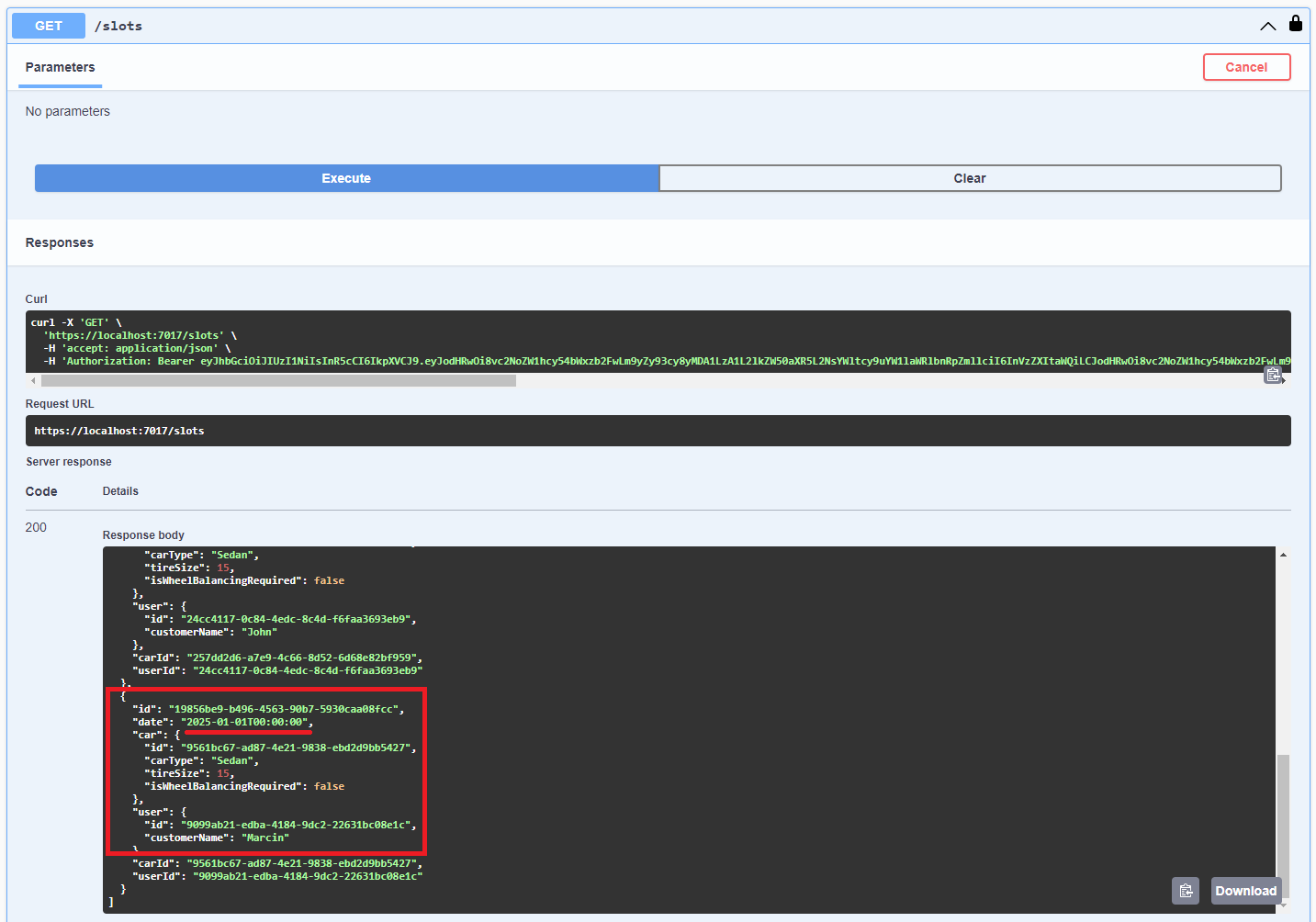
* 1. Authorise



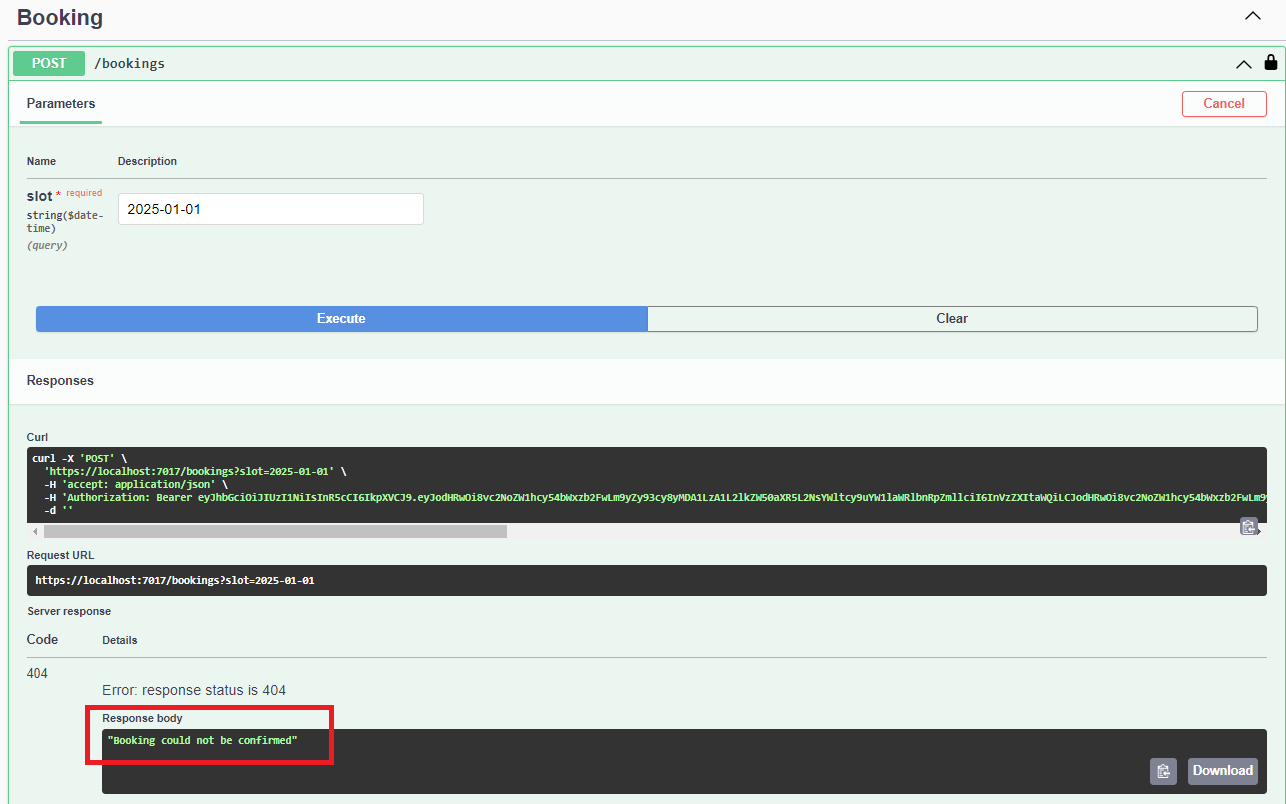
* 1. Get Answer



* 1. Confirm booking



* 1. Try booking a slot when it is booked



1. Caclucate price change tire in a car.

