Java developer exercise

Prepare a simple service, which will let the client store and read payments.

Functional requirements

- Payment is described by: unique identifier created during persistence process, amount, currency, user ID and target bank account number
- Payments should be stored by the service
- Service should expose an API, which should be able to:
  - Fetch payment resources
  - Create, update and delete payment resources
  - List a collection of payment resources

Non-Functional requirements

- API should be RESTFUL
- Application should be able to store payments in a CSV file
- The code should be open for extensions, i.e possibility to add a support for an in-memory database storage, so the type of storage engine could be passed as a configuration parameter
- You should use best practices, for example TDD/BDD, SOLID etc.
- Consider using Clean Architecture or Hexagonal/Ports and Adapters patterns
- Try to simplify your code by using well proven open source frameworks and libraries
- Write the code with production ready quality in mind

Submitting the exercise

Application code should be published on github or gitlab public repository.