**Project: (Requirement)**

**Coding Sample**

Please create a RESTful microservice that implements a card shuffling algorithm, as defined below.  We’d like to see evidence of **test-driven development with unit tests**.  We’d prefer you use Gradle for the build, and Jetty to host, but these aren't requirements.  Use best practices of interfaces and generics for abstraction, preferably implementing a strategy pattern for deploy-time dependency injection of a shuffling algorithm.  Please document your decision making process with comments in the code, especially with regards to any scope reduction.

Requirements:

·         Create a microservice that stores and shuffles card decks.

·         A card may be represented as a simple string such as “5-heart”, or “K-spade”.

·         A deck is an ordered list of 52 standard playing cards.

·         Expose a RESTful interface that allows a user to:

·         PUT an idempotent request for the creation of a new named deck.  New decks are created in some initial sorted order.

·         POST a request to shuffle an existing named deck.

·         GET a list of the current decks persisted in the service.

·         GET a named deck in its current sorted/shuffled order.

·         DELETE a named deck.

·         Design your own data and API structure(s) for the deck.

·         Persist the decks in-memory only, but stub the persistence layer such that it can be later upgraded to a durable datastore.

·         Implement a simple shuffling algorithm that simply randomizes the deck in-place.

·         Implement a more complex algorithm that simulates hand-shuffling, i.e. splitting the deck in half and interleaving the two halves, repeating the process multiple times.

·         Allow switching the algorithms at deploy-time only via configuration.

 Provide the source code and instructions for building/running/using the microservice.

We don't want you to spend more than 4-8 hours on this, so we are interested in seeing your decision making process regarding feature reduction to meet the deadline while creating a minimally viable product. Feel free to ad-lib requirements within the spirit of the exercise as you see fit, or to contact me with any questions about the project.

**System Requirements:**

Java 8

Jetty standalone server

Gradle

Postman

**Solution:**

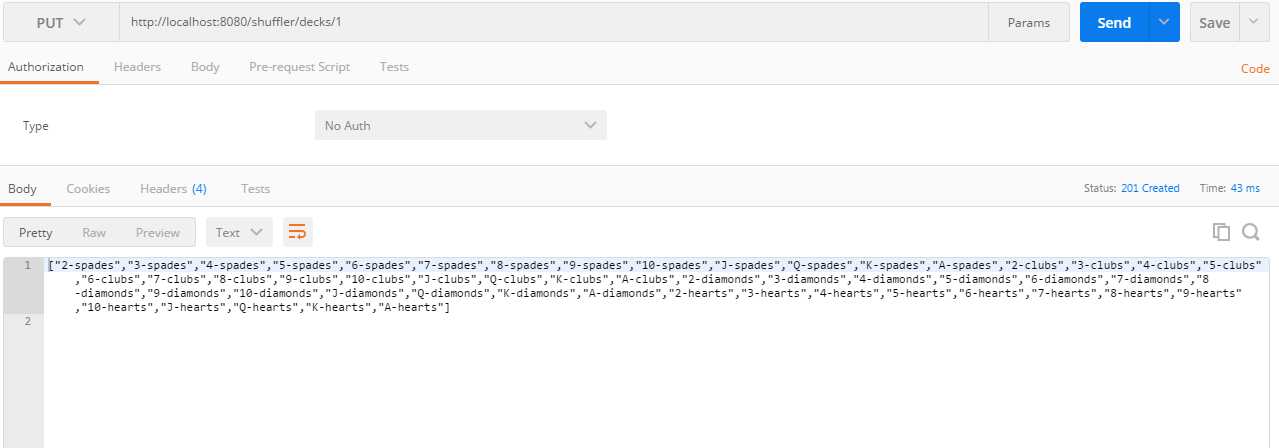
Please look at **Shuffler** folder gives complete project development.

**Test Results:**

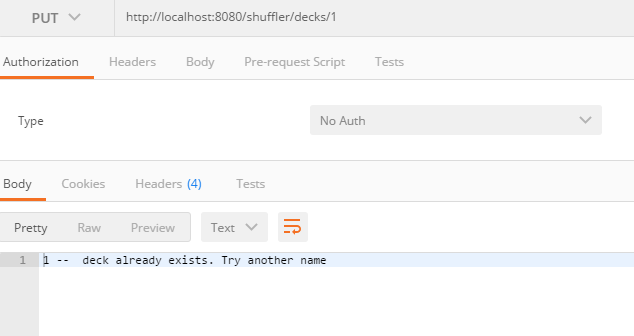
1. **PUT an idempotent request for the creation of a new named deck.  New decks are created in some initial sorted order**

<http://localhost:8080/shuffler/decks/>*{deckName}*

Note: Replace *{deckName}* with the name you want to store the Deck. We shouldn’t provide the existing deck names.

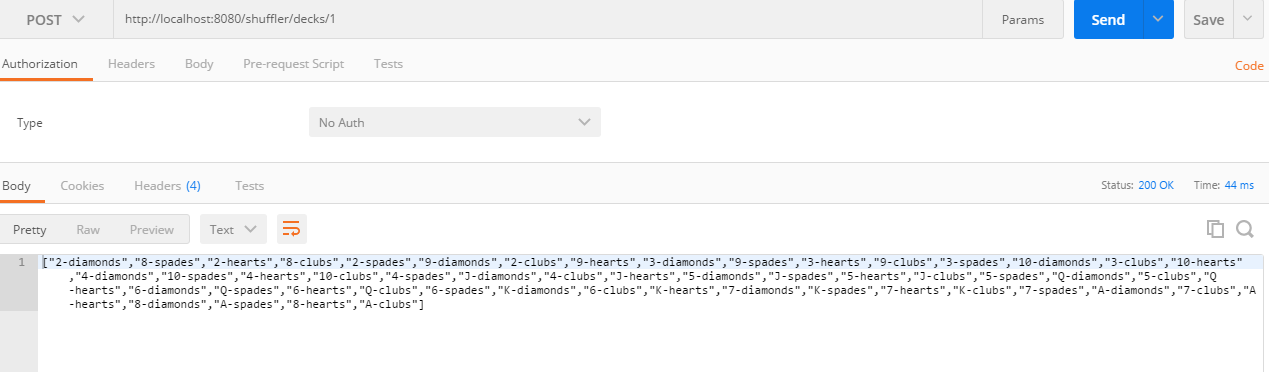


If we provide the same deck name, we get the following error message.

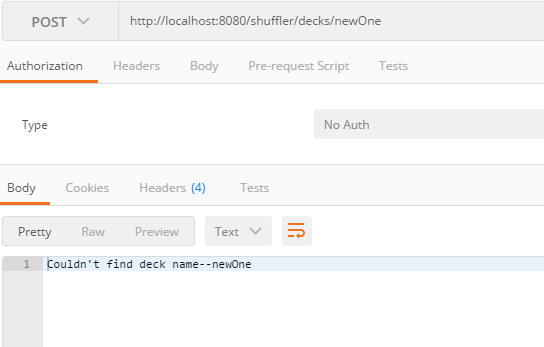


1. **POST a request to shuffle an existing named deck.**

http://localhost:8080/shuffler/decks/{deckName}

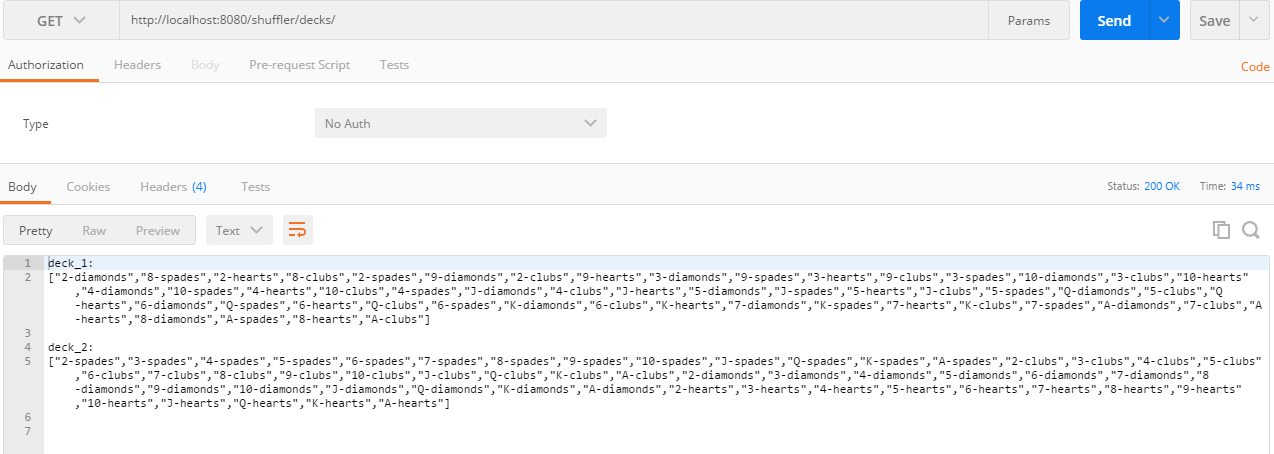


If the name doesn’t exist, the following error message is displayed.



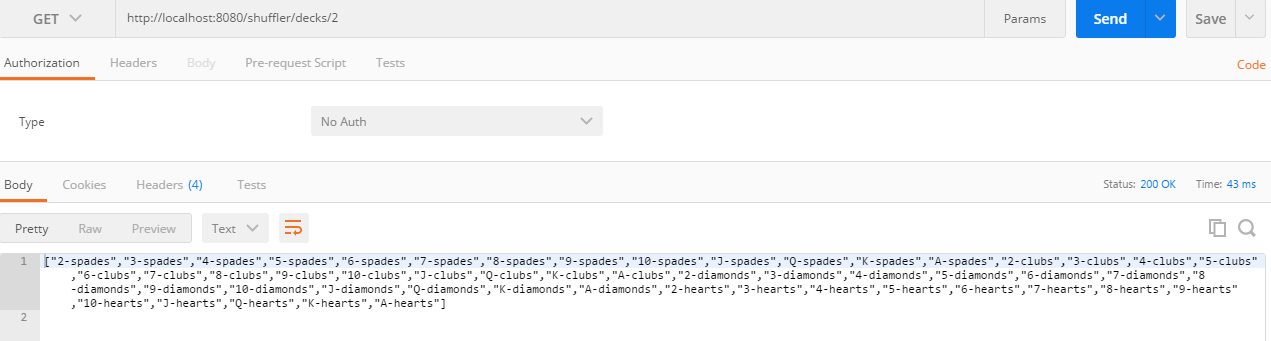
1. **GET a list of the current decks persisted in the service.**

http://localhost:8080/shuffler/decks

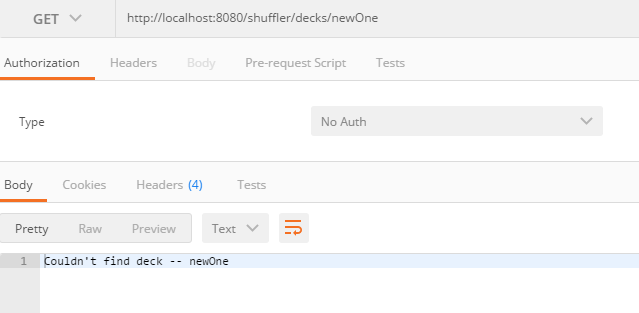


1. **GET a named deck in its current sorted/shuffled order.**

http://localhost:8080/shuffler/decks/{deckName}

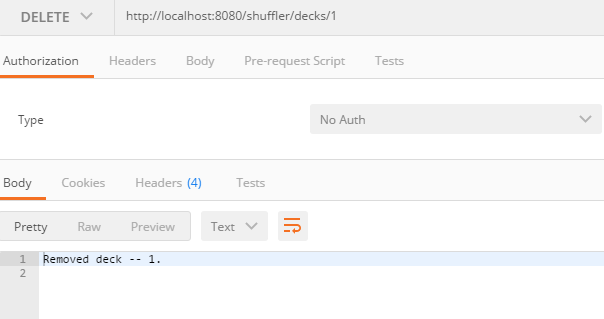


Note: If we provide a deck name that doesn’t exist, we get an error message.



1. **DELETE a named deck.**

http://localhost:8080/shuffler/decks/{deckName}



Once the deck is deleted or deck name doesn’t exist, we get the following error message.

