# Credit Card Processing

Write an API for a credit card provider.

The backend must be a RESTful API. The backend allows you to add new credit card accounts, process charges and credits against them.

We expect the developer to respond to several aspects:

1. Use of modular JavaScript patterns to build the necessary functionality
2. Demonstrate your ability to deliver and enforce high code quality and maintainability – particularly around Single Responsibility, Separation of Concerns and Encapsulation
3. Your knowledge of RESTful API design
4. Your knowledge of testing – writing “refactor friendly” tests that are clear and easy to read
5. We'd like to see how you use Git in your workflow

## Requirements

Four REST Endpoints must be implemented

* "**Add**" will create a new credit card for a given **name**, **card number**, and **limit**
  + Card numbers should be validated using Luhn 10
  + New cards start with a £0 balance
  + for cards not compatible with Luhn 10, return an error
* "**Charge**" will increase the balance of the card associated with the provided **name** by the amount specified
  + Charges that would raise the balance over the limit should not be permitted
  + Respond back with the card number and the remaining balance
* "**Credit**" will decrease the balance of the card associated with the provided **name** by the amount specified
  + Credits that would drop the balance below £0 will create a negative balance
  + Respond back with the card number and the remaining balance
* "**Get all**" returns all cards in the system

The endpoints should be given appropriate URLs and HTTP methods, according to RESTful design principles. There is no right and wrong answer here, but you may be asked to explain and justify your design, so give it some thought.

## Input validation

* all input will be JSON
* credit card numbers may vary in length, up to 19 characters
* credit card numbers will always be numeric
* amounts will always be prefixed with "£" and will be in pounds (Decimal up to 2 places)

## Technical requirements

* Create the endpoints use appropriate HTTP Methods and define the payloads, return codes and response structures
* Factor in all the input validation
* Write unit test cases
* Use an in-memory DB to store the information while the API is running, so that it can store the credit card information and process the charge and credit operations
* Send steps for how to run and test the API
* Hand code the Luhn 10 check, don’t use a library

## Assessment criteria

We will assess your work on the following basis:

* Code structure, quality and consistency
* Orthogonality
* Attention to detail
* Clear documentation
* Test quality
* Git commit history