



# Toggl Backend Unattended Programming Test



## Scenario

You intend to create card games like Poker and Blackjack. The first thing to do would be to create an API to handle the deck and cards to be used in any game like these. So that's what this assignment is all about.

## Instructions

You are required to provide an implementation of a REST API to simulate a deck of cards.

You need to provide a solution written in Go. If you do not feel too comfortable with the language, it's OK to research a little bit before writing your API.

You will need to provide the following methods to your API to handle cards and decks:

- Create a new **Deck**
- Open a **Deck**
- Draw a **Card**

## Background

### | Create a new **Deck**

It would create the standard 52-card deck of French playing cards, It includes all thirteen ranks in each of the four suits: clubs (♣), diamonds (♦), hearts (♥) and spades (♠). You don't need to worry about Joker cards for this assignment.

You should allow the following options to the request:

- the deck to be shuffled or not — by default the deck is sequential: A-spades, 2-spades, 3-spades... followed by diamonds, clubs, then hearts.
- the deck to be full or partial — by default it returns the standard 52 cards, otherwise the request would accept the wanted cards like this example

```
?cards=AS,KD,AC,2C,KH
```

The response needs to return a JSON that would include:

- the deck id (**UUID**)
- the deck properties like shuffled (**boolean**) and total cards remaining in this deck (**integer**)

```
{
  "deck_id": "a251071b-662f-44b6-ba11-e24863039c59",
  "shuffled": false,
  "remaining": 30
}
```

## Open a **Deck**

It would return a given deck by its UUID. If the deck was not passed over or is invalid it should return an error. This method will "open the deck", meaning that it will list all cards by the order it was created.

The response needs to return a JSON that would include:

- the deck id (**UUID**)
- the deck properties like shuffled (**boolean**) and total cards remaining in this deck (**integer**)
- all the remaining cards cards (**card object**)

```
{
  "deck_id": "a251071b-662f-44b6-ba11-e24863039c59",
  "shuffled": false,
  "remaining": 3,
  "cards": [
    {
      "value": "ACE",
      "suit": "SPADES",
      "code": "AS"
    },
    {
      "value": "KING",
      "suit": "HEARTS",
      "code": "KH"
    },
    {
      "value": "8",
      "suit": "CLUBS",
      "code": "8C"
    }
  ]
}
```

## Draw a **Card**

I would draw a card(s) of a given Deck. If the deck was not passed over or invalid it should return an error. A count parameter needs to be provided to define how many cards to draw from the deck.

The response needs to return a JSON that would include:

- all the drawn cards cards (**card object**)

```
{
  "cards": [
    {
      "value": "QUEEN",
      "suit": "HEARTS",
      "code": "QH"
    },
    {
      "value": "4",
      "suit": "DIAMONDS",
      "code": "4D"
    }
  ]
}
```

## What we are looking for

We are interested in:

- How you structure your code do that and it's
  - well tested
  - easy to extend (think about the other card games you want to create)
  - easy to modify
  - easy to understand to others
  - compiles with best Go practices
- Evidence of your of Back-end development knowledge
- Evidence of testing (TDD, BDD)

We do not expect a polished solution for this test, so please do not spend time working outside the required scope of this exercise.

Please supply us with your source code, a README file that explains how to build and run your submission, build scripts and any tests you have written.

We expect this test to take approximately 3 hours.