Your mission if you choose to accept it, is to tokenize Credit Cards. It seems like a simple manner, but we spiced it up a little bit, so you would find it more interesting. The CC numbers will self destruct in a few seconds... so have fun.

If you have any questions you may WhatsApp itai @ 050-2625698 or email itai@forter.com;erez@forter.com;oren@forter.com .

The IDE we use here for Nodejs is WebStorm which is free for 30 days.

Credit Card Server

Write a nodejs server listening on port 8081 that implements RESTful API.

The following linux command lines represents a RESTful client inserting a new credit card:

```
curl -X POST -H "Content-Type: application/json"
http://localhost:8081/creditcard -d '{"credit-card":"1234-5678-9101-
1121"}'
```

The response is a unique id that represents the credit card token (you can google "nodejs uuid"):

```
{"token" : "71bf9487-9aeb-4252-a473-2f0efdc709e3" }
```

The following linux command line represents a RESTful client retrieving a stored credit card.

```
curl -X GET http://localhost:8081/creditcard/71bf9487-9aeb-4252-a473-
2f0efdc709e3
```

The result should be the origin credit card:

```
{"credit-card":"1234-5678-9101-1121"}
```

```
For this part of the exercise you can store the credit cards in a
nodejs object and each UUID is a key, and each credit card is the
value.

$ nodejs

> storage = {}

{}

> storage["71bf9487-9aeb-4252-a473-2f0efdc709e3"]="1234-5678-9101-
1121"

'1234-5678-9101-1121'
```

```
> storage["71bf9487-9aeb-4252-a473-2f0efdc709e3"]
'1234-5678-9101-1121'
```

Authenticated Credit Card Server

Run nginx process (on the same machine) that listens on port 80, performs basic authentication (constant username, constant password) and forwards the request to the nodejs server on port 8081.

Bonus points: Asynchronous storage

In this part you would need to store the credit card on an external storage service instead of a nodejs object. This requires understanding async programming in nodejs. For this purpose you should install redis (http://redis.io/download ) which is the simplest storage service, and use the node_redis library.

Bonus points: Validate Credit Card format

Validate the input using z-schema (https://github.com/zaggino/z-schema ) and the luhn algorithm (https://github.com/JamesEggers1/node-luhn)

Bonus points: Amazon Web Sevices EC2

As a bonus you could install everything on an ec2 machine and open port 80 so we can access the result from our own laptops. Make sure that you remember to stop the machine instance on ec2 since it is being billed by the hour. Also note that terminating the machine will delete all of its content while stopping the machine will only delete /mnt (ephemeral storage) and will not delete the / drive (EBS storage).

Good Luck.