

## Instrucciones

Swagger: <https://bank-app-392505.uc.r.appspot.com/swagger-ui/index.html>

En este enlace se encuentran listados los diferentes endpoints de la app.

### Métodos:

GET -> **/card/{productId}/number** : Este método valida que no se repita el número aleatorio que genera, el cual es conformado por un número random de 10 dígitos añadido posterior al *productId* recibido. Finalmente guarda la tarjeta con el número que se generó.

POST -> **/card/enroll** : Este método valida que exista la tarjeta con el número que se envía en el body, luego de esto añade la información del propietario, la fecha de vencimiento y se activa la tarjeta con su saldo inicial. Si no genera un error.

DELETE -> **/card/{cardId}** : Este método valida que la tarjeta exista y luego pasa a cambiar el estado de la tarjeta a bloqueado. Si no genera un error.

POST -> **/card/balance** : Este método valida que la tarjeta exista y luego de ello suma a el saldo actual el nuevo valor recibido. Si no genera un error.

GET -> **/card/balance/{cardId}** : Este método valida que exista la tarjeta con el número indicado y devuelve su saldo. Si no genera un error.

POST -> **/transaction/purchase** : Este método realiza una serie de validaciones, primero valida que exista la tarjeta indicada, luego que esté activa y finalmente que el saldo alcance para la transacción que se desea y descuenta este del saldo de la respectiva tarjeta. Si no cumple con alguna validación genera su debido mensaje de error.

GET -> **/transaction/{transactionId}** : Este método valida que exista la transacción y posteriormente devuelve toda su información, de lo contrario genera el respectivo error.

POST -> **/transaction/anulation** : Este método valida que exista la transacción, que no se haya realizado un día antes, cambia el estado a la transacción y devuelve el saldo a la respectiva tarjeta. Cada validación devuelve su mensaje de error.

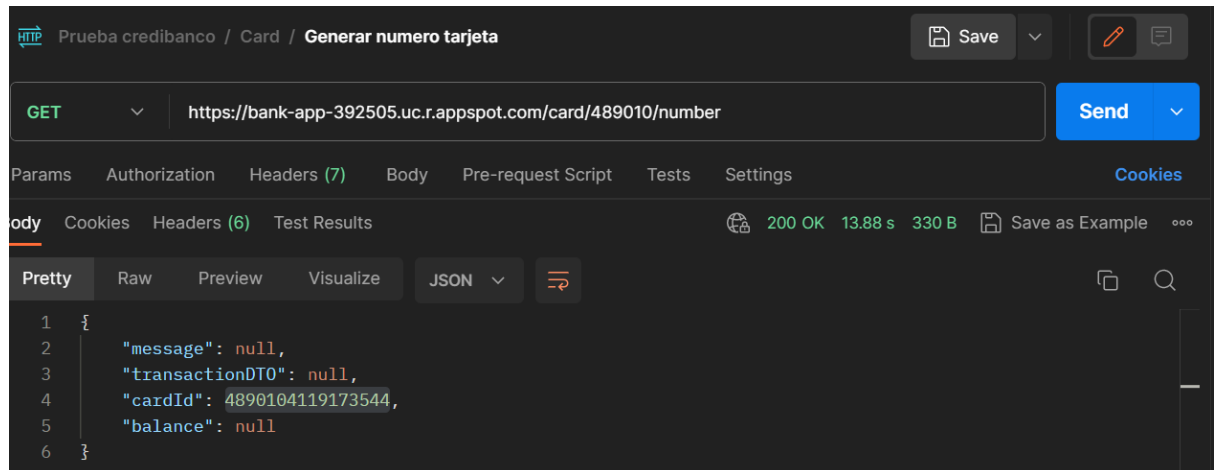
GET -> **/transaction** : Este método se realizó para observar todas las tarjetas registradas.

GET -> **/card** : Este método se realizó para observar todas las transacciones registradas.

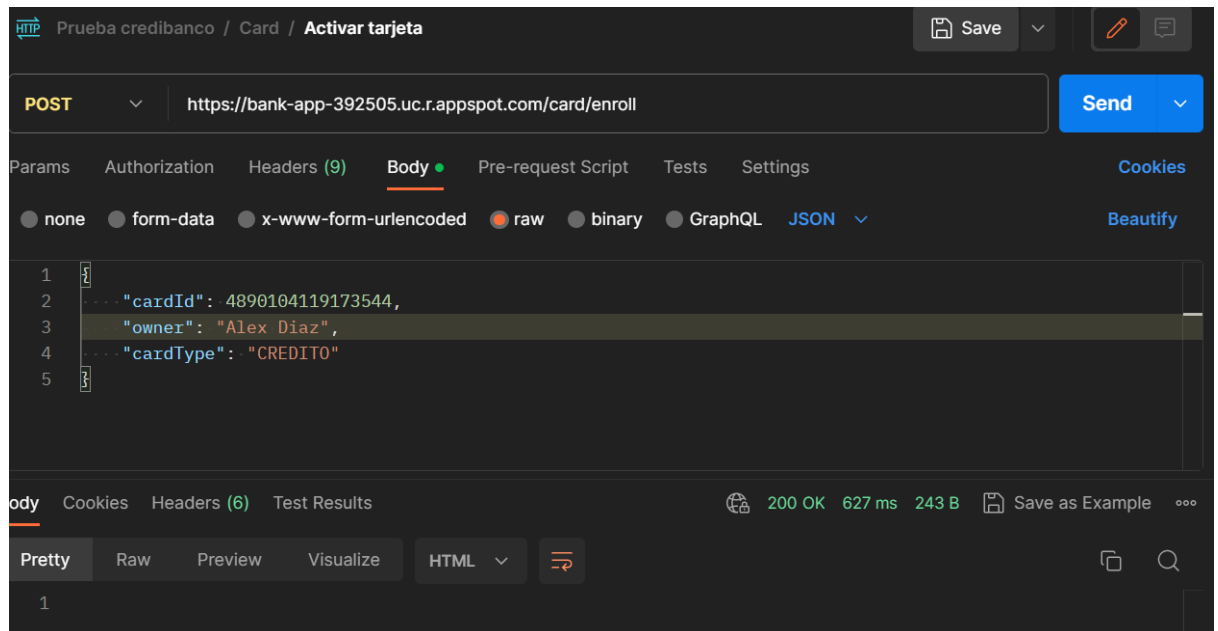
### Pruebas con postman:

Un ciclo de prueba realizado fue el siguiente:

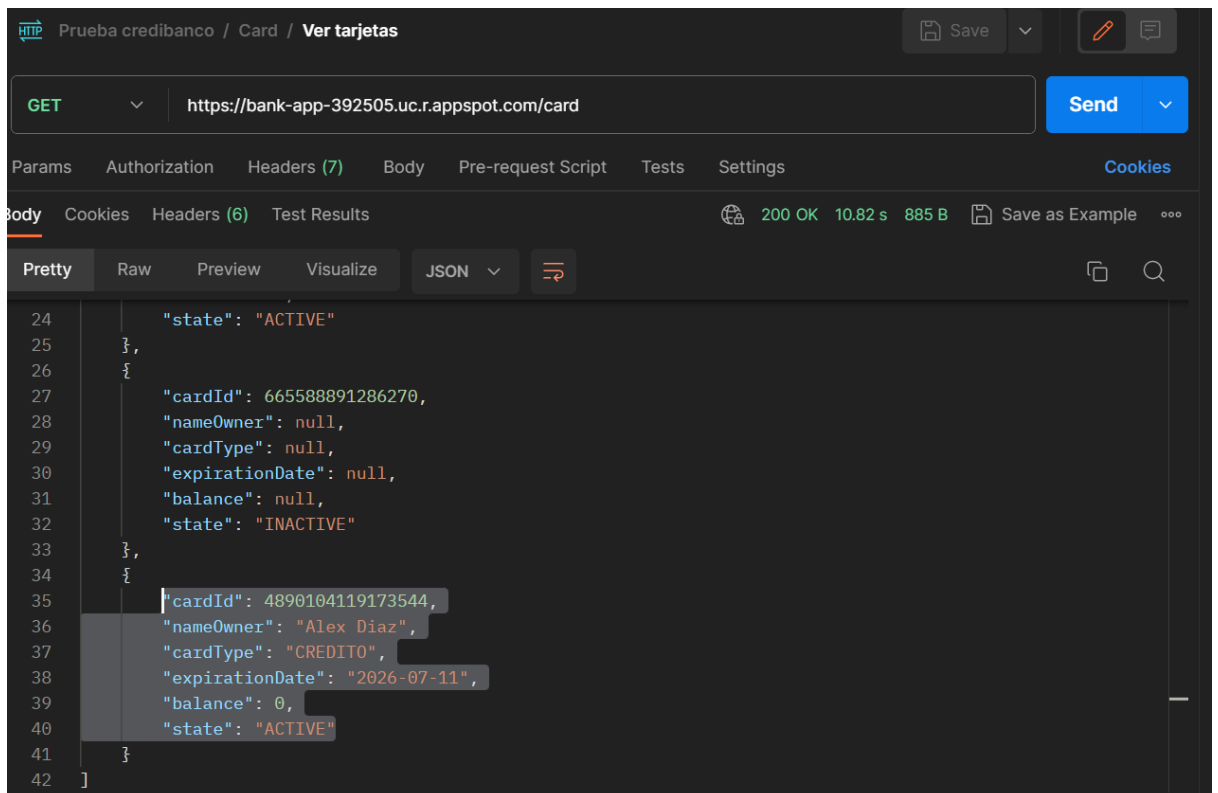
1. Se genera el número de tarjeta **4890104119173544**



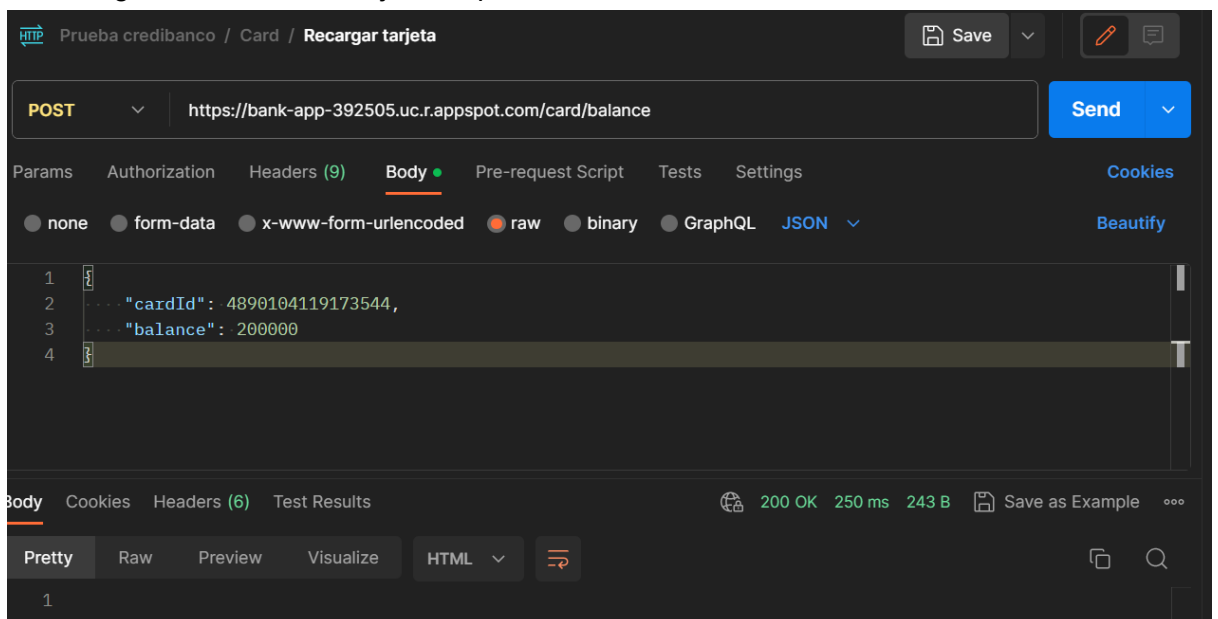
2. Se activa la tarjeta **4890104119173544** a nombre de **Alex Diaz** de tipo **CREDITO**



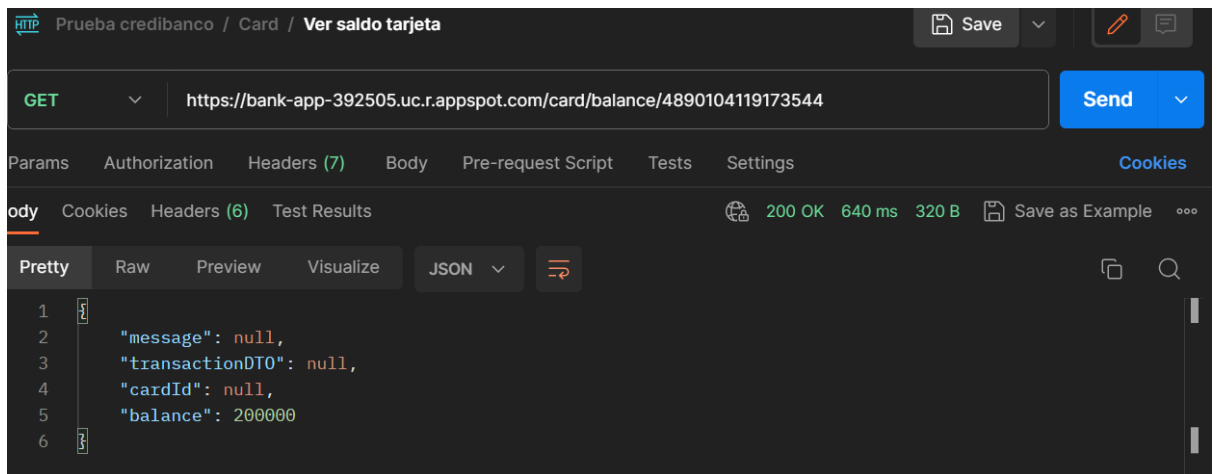
Se valida la tarjeta



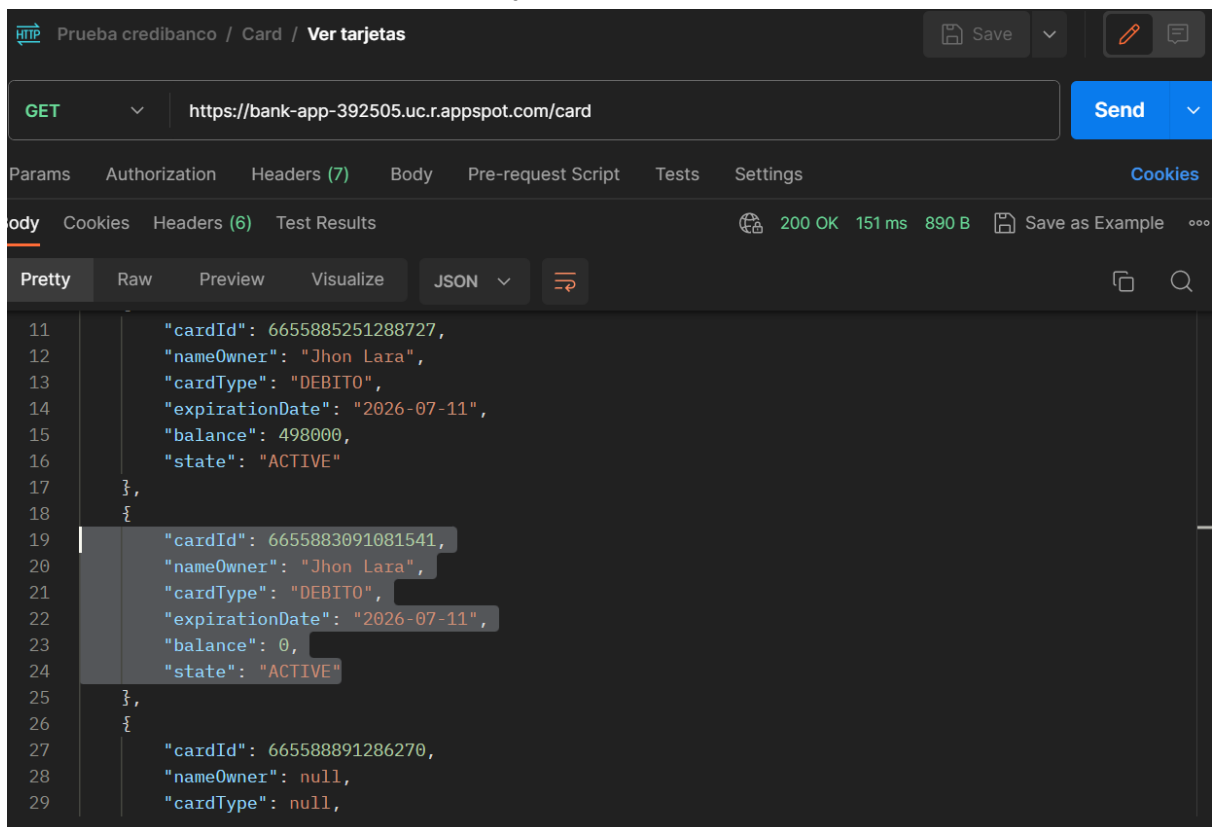
3. Se recarga con 200000 la tarjeta de prueba.



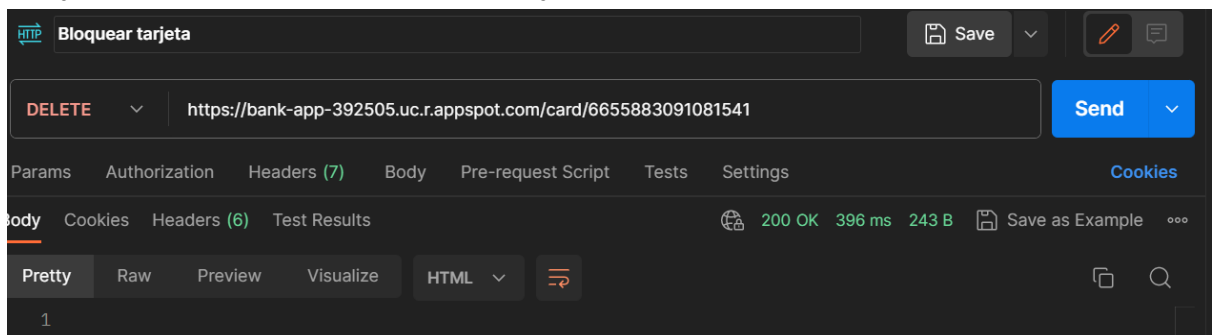
4. Se valida el saldo con el método creado para esto.



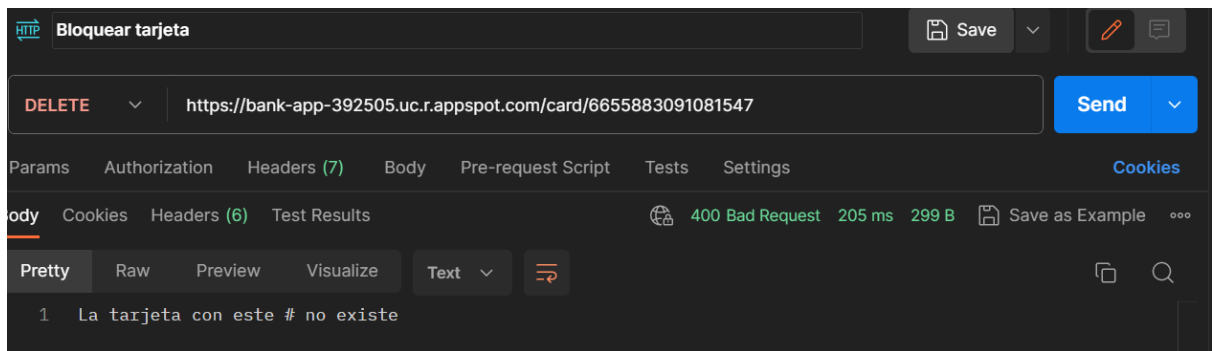
5. Se valida que esté activa una de las tarjetas.



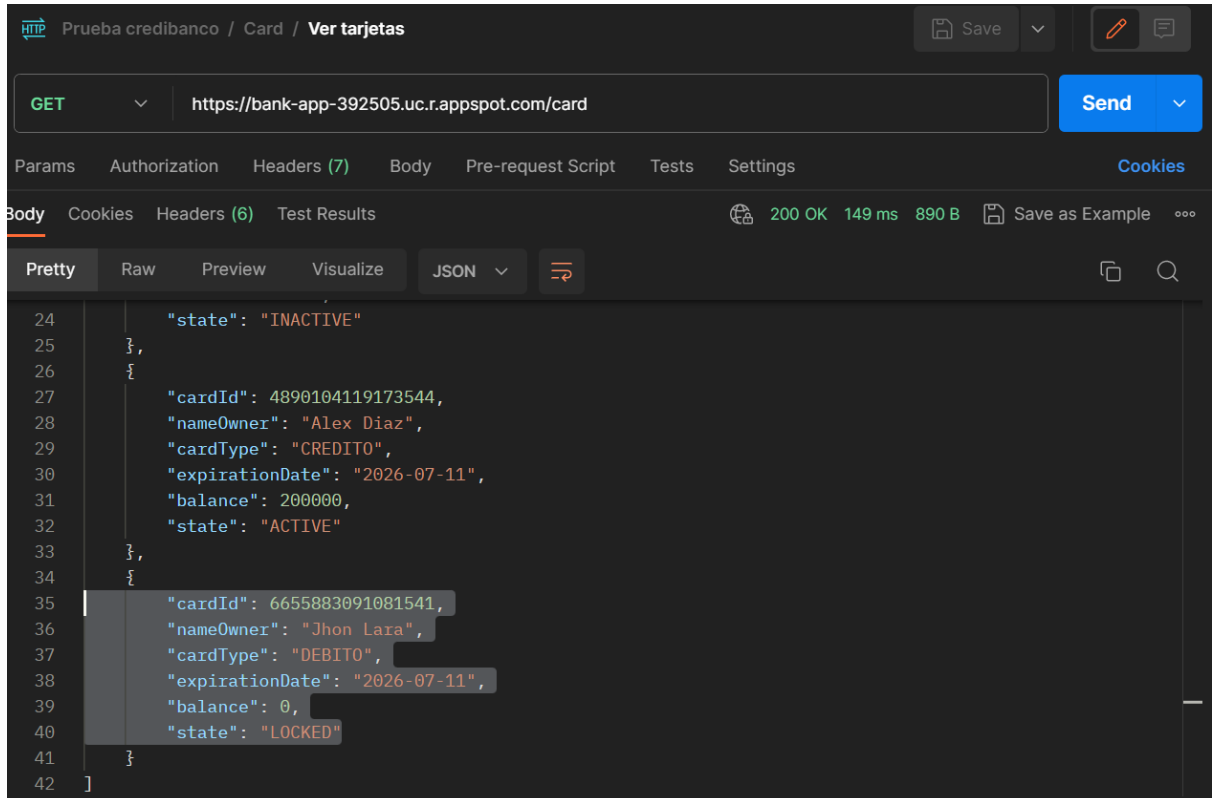
Se ejecuta el servicio para bloquear la tarjeta.



Acá podemos notar que al poner un número no existente el error aparece.



Se valida nuevamente la anterior ejecución.



6. Se valida anterior saldo

```
GET https://bank-app-392505.uc.r.appspot.com/card

200 OK 152 ms 890 B

[{"state": "INACTIVE", "cardId": 6655883091081541, "nameOwner": "Jhon Lara", "cardType": "DEBITO", "expirationDate": "2026-07-11", "balance": 0, "state": "LOCKED"}, {"cardId": 4890104119173544, "nameOwner": "Alex Diaz", "cardType": "CREDITO", "expirationDate": "2026-07-11", "balance": 199000, "state": "ACTIVE"}]
```

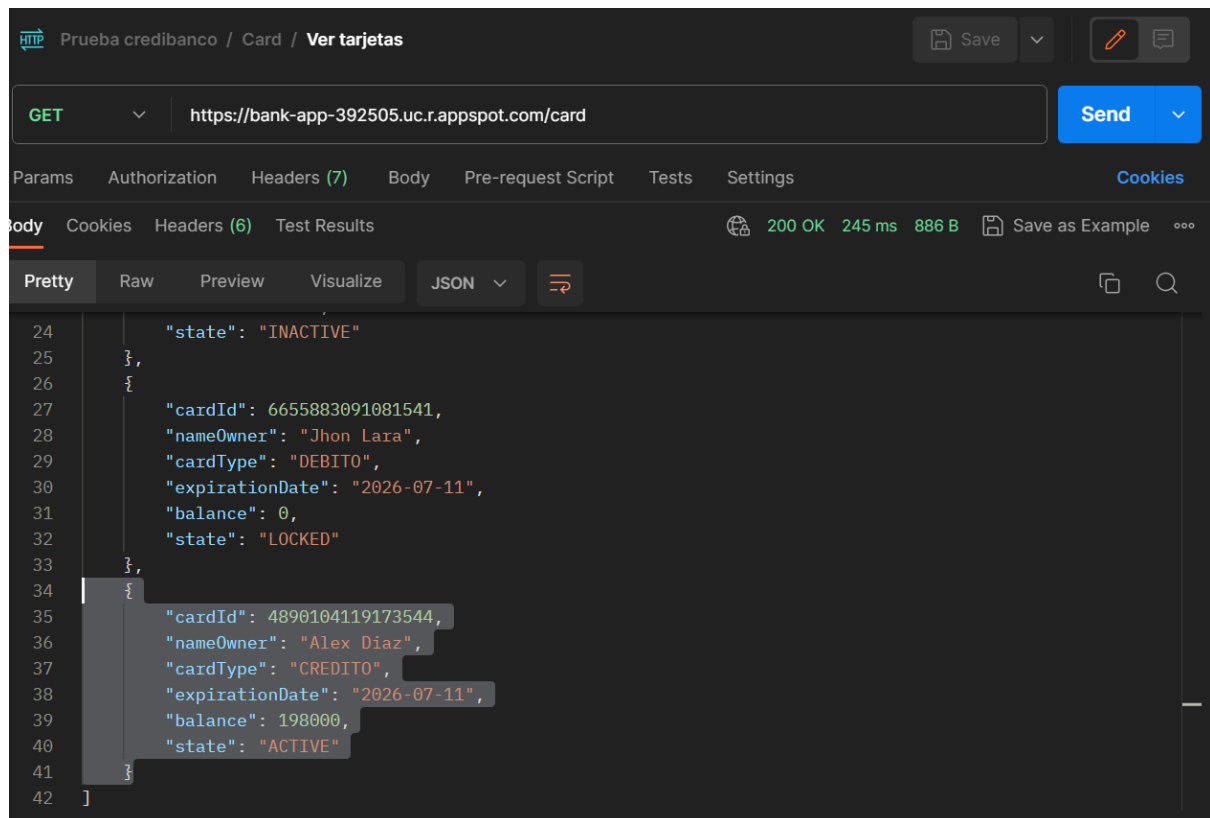
Se descuenta **1000** del saldo **19900**.

```
POST https://bank-app-392505.uc.r.appspot.com/transaction/purchase

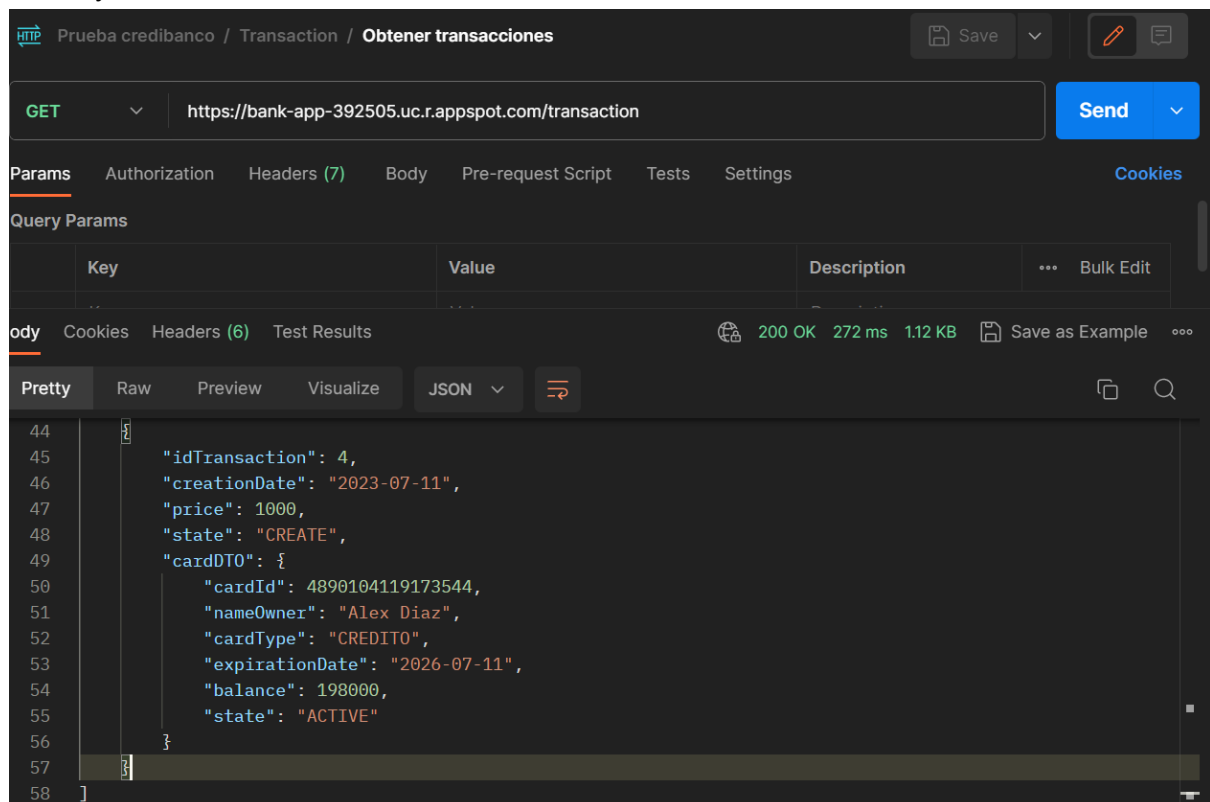
{"cardId": 4890104119173544, "balance": 1000}

200 OK 674 ms 243 B
```

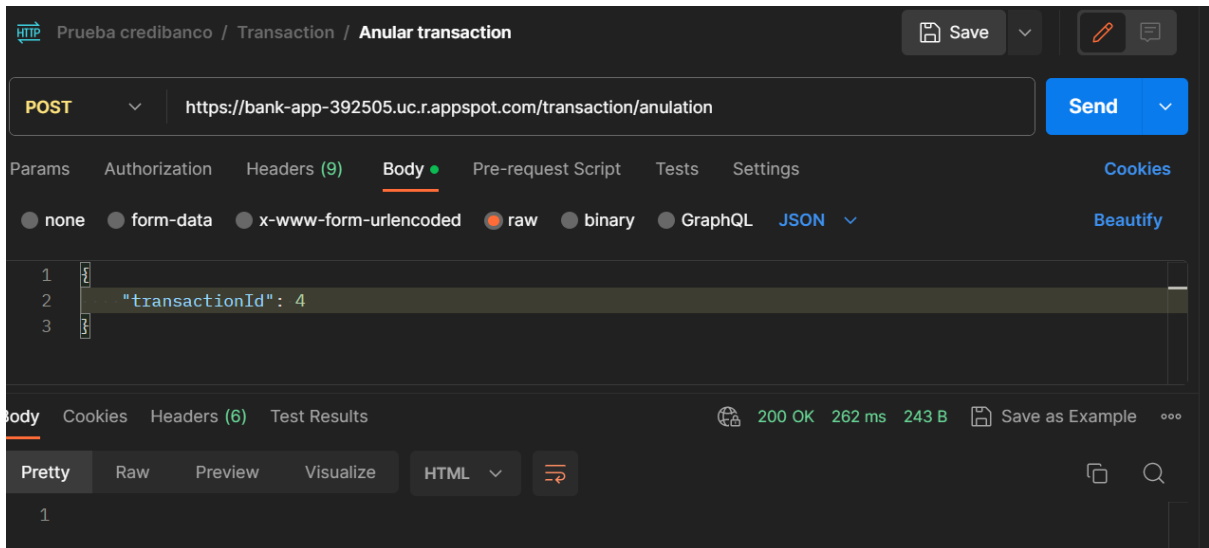
Se valida que quede **198000**.



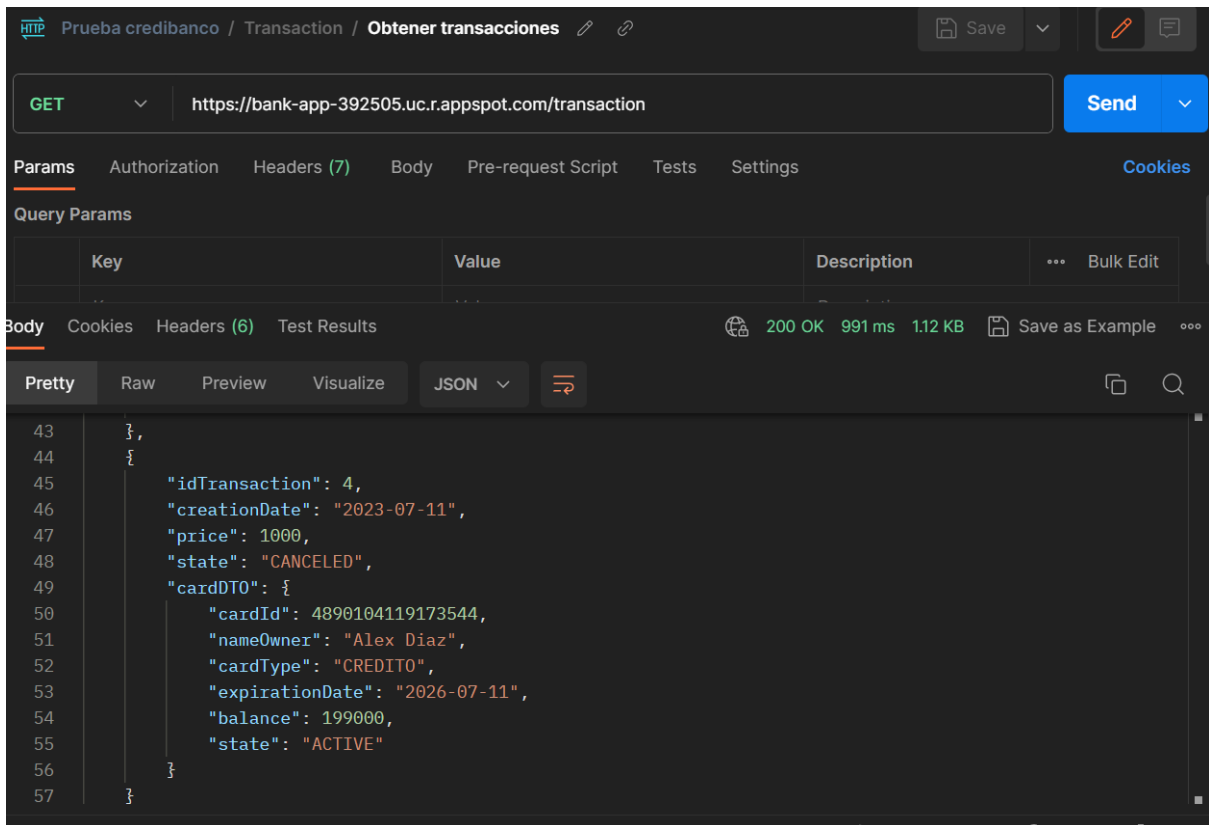
7. Se valida la transacción 4. La cual tiene un saldo de **198000**, luego del movimiento de **10000** y tiene estado **ACTIVO**.



Se valida la actualización



Con el nuevo saldo y el nuevo estado.



8. Se valida el servicio que trae una transacción con id 4.



Prueba credibanco / Transaction / Obtener transaction

GET https://bank-app-392505.uc.r.appspot.com/transaction/4 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

body Cookies Headers (6) Test Results 200 OK 214 ms 541 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": null,
3   "transactionDTO": {
4     "idTransaction": 4,
5     "creationDate": "2023-07-11",
6     "price": 1000,
7     "state": "CANCELED",
8     "cardDTO": {
9       "cardId": 4890104119173544,
10      "nameOwner": "Alex Diaz",
11      "cardType": "CREDITO",
12      "expirationDate": "2026-07-11",
13      "balance": 199000,
14      "state": "ACTIVE"
15    }
16  },
17   "cardId": null,
18   "balance": null
19 }
```