

API Integración para D-TPV 2

1. Comunicación.....	3
1.1 Explicación de comunicación.....	3
1.2 Seguridad en la comunicación.....	3
2. Consultas.....	4
2.1 Consulta de saldo.....	4
2.2 Consulta de movimientos.....	5
2.3 Recarga de saldo.....	5
2.4 Consumo de saldo.....	6
2.5 Creación / asignación de tarjeta.....	6
2.6 Eliminación / desasignación de tarjeta.....	7
2.7 Creación de factura para TicketBAI/Batus.....	8
3. Comandos.....	9
3.1 Leer tarjeta para realizar recarga.....	9
3.2 Leer tarjeta para realizar consumo.....	10

1. Comunicación

1.1 Explicación de comunicación.

La comunicación se realiza mediante archivos en formato **JSON**. Los archivos para realizar la consulta se deben guardar en la carpeta **c:\infodtpv2\com\in** con extensión **.txt**. con el nombre que se desee, el sistema cada segundo comprobará si existen consultas en dicha carpeta y tras analizar la consulta borrara el archivo de la carpeta **c:\infodtpv2\com\in** y dejará la respuesta en la carpeta **c:\infodtpv2\com\out** con el mismo nombre de archivo.

Ejemplo:

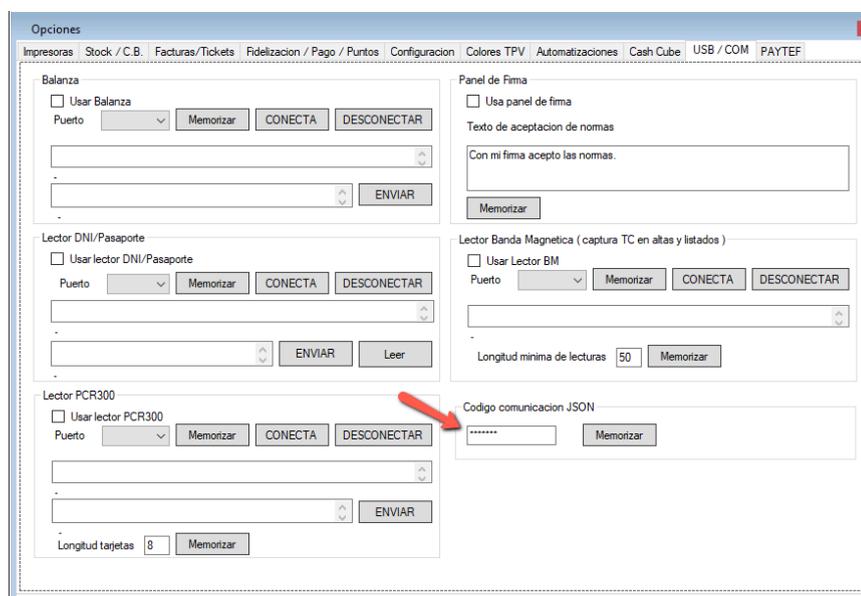
Se deja archivo llamado 12122022-121248.txt en la carpeta **c:\infodtpv2\com\in**

el sistema lee el archivo, lo elimina y la respuesta la deja en un archivo llamado 12122022-121248.txt en la carpeta **c:\infodtpv2\com\out**.

Se recomienda que el archivo de respuesta se borre al tratar el mismo.

1.2 Seguridad en la comunicación

Cada consulta debe ser enviada con el código de comunicación definido en el programa en la ventana de Opciones, pestaña USB/COM, de lo contrario las consultas serán rechazadas.



2. Consultas

2.1 Consulta de saldo

Datos a enviar en el archivo JSON:

type: 1 (Tipo de operación de consulta de saldo)

card: Numero de tarjeta del cliente

apikey: Código de comunicación

Ejemplo consulta:

```
{"type":"1","card":"1234567890","apikey":"codecom"}
```

Respuesta:

type: 1 (confirmación de tipo de operación)

card: Número de tarjeta del cliente (confirmación de número de tarjeta de cliente)

name: nombre al que está asociada la tarjeta

credit: saldo de la tarjeta

Ejemplo respuesta:

```
{"type":"1","card":"1234567890","name":"nombre del cliente","credit":"12,96"}
```

2.2 Consulta de movimientos

Datos a enviar en el archivo JSON:

type: 2 (Tipo de operación de consulta movimientos)

card: Numero de tarjeta del cliente

date_from: Fecha desde donde se realiza la consulta. Formato (YYYY-MM-DD)

date_to: Fecha hasta donde se realiza la consulta. Formato (YYYY-MM-DD)

apikey: Código de comunicación

Ejemplo consulta:

```
{"type":"1","card":"1234567890","date_from":"2022-10-01","date_to":"2022-10-31","apikey":"codecom"}
```

Respuesta:

type: 2 (confirmación de tipo de operación)

card: Número de tarjeta del cliente (confirmación de número de tarjeta de cliente)
date: Fecha de operación
op_type: Tipo de operación (Venta o Recarga)
amount: Importe de la operación
desc: Descripción de la operación

Ejemplo respuesta:

```
[{"type":"2","card":"1234567890","date":"2022-10-01  
14:12:43","op_type":"Recarga","amount":"50","desc":"Recarga de  
saldo"}, {"type":"2","card":"1234567890","date":"2022-10-01  
14:16:12","op_type":"Compra","amount":"12.50","desc":"Consumición  
bar"}, {"type":"2","card":"1234567890","date":"2022-10-01  
18:04:37","op_type":"Compra","amount":"8.70","desc":"Consumición piscina"}]
```

2.3 Recarga de saldo

Datos a enviar en el archivo JSON:

type: 3 (Tipo de operación de recarga de saldo)
card: Numero de tarjeta del cliente
amount: Importe a recargar (Formato double con punto como separador decimal) que se sumará al saldo actual de la tarjeta
desc: Descripción de la operación
apikey: Código de comunicación

Ejemplo consulta:

```
{"type":"3","card":"1234567890","amount":"50.50","desc":"Recarga en  
recepcion","apikey":"codecom"}
```

Respuesta:

type: 3 (confirmación de tipo de operación)
card: Número de tarjeta del cliente (confirmación de número de tarjeta de cliente)
credit: nuevo saldo de la tarjeta.

Ejemplo respuesta:

```
{"type":"3","card":"1234567890","credit":"50,50"}
```

2.4 Consumo de saldo

Datos a enviar en el archivo JSON:

type: 4 (Tipo de operación de consumo de saldo)

card: Numero de tarjeta del cliente

amount: Importe a descontar (Formato double con punto como separador decimal)

desc: Descripción de la operación

apikey: Código de comunicación

Ejemplo consulta:

```
{"type": "4", "card": "1234567890", "amount": "20.50", "desc": "Consumición de terraza", "apikey": "codecom"}
```

Respuesta:

type: 4 (confirmación de tipo de operación)

card: Número de tarjeta del cliente (confirmación de número de tarjeta de cliente)

error: 0 si la operación se realizó correctamente, descripción de error si no se realizó correctamente.

credit: nuevo saldo de la tarjeta

Ejemplo respuesta si la operacion se realizo correctamente:

```
{"type": "4", "card": "1234567890", "error": "0", "credit": "10.50"}
```

Ejemplo respuesta si la operación no se realizó correctamente por falta de saldo:

```
{"type": "4", "card": "1234567890", "error": "Saldo insuficiente", "credit": "1.50"}
```

2.5 Creación / asignación de tarjeta

Datos a enviar en el archivo JSON:

type: 5 (Tipo de operación de creación de tarjeta)

card: Numero de tarjeta del cliente

amount: Importe a recargar (Formato double con punto como separador decimal, puede ser cero)

name: nombre del cliente

apikey: Código de comunicación

Ejemplo consulta:

```
{"type": "5", "card": "1234567890", "amount": "50", "name": "nombre del cliente", "apikey": "codecom"}
```

Respuesta:

type: 5 (confirmación de tipo de operación)

card: Número de tarjeta del cliente (confirmación de número de tarjeta de cliente)

error: 0 si la operación se realizó correctamente, descripción de error si no se realizó correctamente.

Ejemplo respuesta exitosa:

```
{"type": "5", "card": "1234567890", "error": "0"}
```

Ejemplo respuesta no exitosa:

```
{"type": "5", "card": "1234567890", "error": "Esa tarjeta ya está asociada a otro cliente"}
```

2.6 Eliminación / desasignación de tarjeta

Datos a enviar en el archivo JSON:

type: 6 (Tipo de operación de eliminación de tarjeta)

card: Numero de tarjeta del cliente

apikey: Código de comunicación

Ejemplo consulta:

```
{"type": "6", "card": "1234567890", "apikey": "codecom"}
```

Respuesta:

type: 6 (confirmación de tipo de operación)

card: Número de tarjeta del cliente (confirmación de número de tarjeta de cliente)

error: 0 si la operación se realizó correctamente, descripción de error si no se realizó correctamente.

Ejemplo respuesta exitosa:

```
{"type": "6", "card": "1234567890", "error": "0"}
```

Ejemplo respuesta no exitosa:

```
{"type": "6", "card": "1234567890", "error": "No existe cliente con ese número de tarjeta"}
```

2.7 Creación de factura para TicketBAI/Batuz

Datos a enviar en el archivo JSON:

type: 9 (Tipo de operación de creación de factura TicketBAI/Batuz)

destinatario_nif: Nif del cliente

destinatario_apellidosnombrazonsocial: Apellidos, Nombre / Razón social

seriefactura: Serie de la factura

numfactura: Número de la factura

fechaexpedicionfactura: Fecha de la expedición de la factura

horaexpedicionfactura: Hora de la expedición de la factura

descripcionfactura: Descripción de la factura

importetotalfactura: Importe total de la factura

baseimponible: Base imponible de la factura

tipويمpositivo: Tipo de IVA de la factura

cuotaimpuesto: Cuota de IVA de la factura

apikey: Código de comunicación

Ejemplo consulta:

```
{"type": "9", "destinatario_nif": "B123456789", "destinatario_apellidosnombrazonsocial": "Empresa de muestra", "seriefactura": "A2024", "numfactura": "3399", "fechaexpedicionfactura": "2024-01-15", "horaexpedicionfactura": "12:14:56", "descripcionfactura": "descripcion de factura", "importetotalfactura": "31.46", "baseimponible": "26.00", "tipoimpositivo": "21", "cuotaimpuesto": "5.46", "apikey": "codecom"}
```

Respuesta:

type: 9 (confirmación de tipo de operación)

seriefactura: Serie de la factura

numfactura: Número de la factura

codqr: código del QR de Ticketbai generado

error: 0 si la operación se realizó correctamente, descripción de error si no se realizó correctamente.

Ejemplo respuesta exitosa:

```
{"type": "9", "seriefactura": "A2024", "numfactura": "3399", "codqr": "TBAI-B13795257-020424-b4U+aATwCRqI+-224", "error": "0"}
```

Ejemplo respuesta no exitosa:

```
{"type": "9", "seriefactura": "A2024", "numfactura": "3399", "codqr": "", "error": "(B4_2000033)La Cuota del impuesto tiene un valor incorrecto para los campos Tipo impositivo y Base imponible suministrados."}
```

Si se selecciona crear la imagen del de QR en lugar de imprimir la FRA/Ticket esta se guardará en un archivo con el mismo nombre del JSON con la extensión **.jpg** en la carpeta **out**.

3. Comandos

3.1 Leer tarjeta para realizar recarga

Orden para que el programa saque la ventana de lectura de tarjeta, una vez leída la tarjeta mostrará la ventana de recarga para seleccionar importe y descripción de la operación. Tras completarse la operación se devolverá un JSON con el resultado de la misma.

Datos a enviar en el archivo JSON:

type: 7 (Tipo de operación de lectura de tarjeta por parte del software para recarga)

apikey: Código de comunicación

Ejemplo consulta:

```
{"type": "7", "apikey": "codecom"}
```

Respuesta:

type: 7 (confirmación de tipo de operación)

card: Número de tarjeta del cliente (confirmación de número de tarjeta de cliente)

desc: descripción de la operación

amount: Importe recargado que se sumó al saldo de la tarjeta.

error: 0 si la operación se realizó correctamente, descripción de error si no se realizó correctamente.

credit: nuevo saldo de la tarjeta.

Ejemplo respuesta si se realizó la recarga correctamente:

```
{"type": "7", "card": "1234567890", "desc": "Recarga manual", "amount": "40", "error": "0", "credit": "50.50"}
```

Ejemplo respuesta si no realizó la recarga correctamente por diversos motivos:

```
{"type": "7", "card": "1234567890", "desc": "", "amount": "", "error": "No existe la tarjeta", "credit": ""}
```

```
{"type": "7", "card": "1234567890", "desc": "", "amount": "", "error": "Operación cancelada", "credit": ""}
```

3.2 Leer tarjeta para realizar consumo

Orden para que el programa saque la ventana de lectura de tarjeta, una vez leída la tarjeta mostrará la ventana de cobro para seleccionar importe y descripción de la operación. Tras completarse la operación se devolverá un JSON con el resultado de la misma.

Datos a enviar en el archivo JSON:

type: 8 (Tipo de operación de lectura de tarjeta por parte del software para consumo)

apikey: Código de comunicación

Ejemplo consulta:

```
{"type": "8", "apikey": "codecom"}
```

Respuesta:

type: 8 (confirmación de tipo de operación)

card: Número de tarjeta del cliente (confirmación de número de tarjeta de cliente)

desc: descripción de la operación

amount: Importe consumido que se descontó al saldo de la tarjeta.

error: 0 si la operación se realizó correctamente, descripción de error si no se realizó correctamente.

credit: nuevo saldo de la tarjeta.

Ejemplo respuesta si se realizó la recarga correctamente:

```
{"type":"8","card":"1234567890","desc":"Consumición en bar","amount":"7.50","error":"0","credit":"40.50"}
```

Ejemplo respuesta si no realizo la recarga correctamente por diversos motivos:

```
{"type":"8","card":"1234567890","desc":"","amount":"","error":"No existe la tarjeta","credit":""}
```

```
{"type":"8","card":"1234567890","desc":"","amount":"","error":"Operación cancelada","credit":""}
```

Revisor	Rev.	Fecha	Notas
Serafin moral	1.00	12/11/2022	Creación de API
Serafin moral	1.01	15/01/2024	Inclusión de funciones para integración con TicketBAI/Batuz
Serafin moral	1.02	04/04/2024	Inclusión de imagen de QR y texto TicketBai/Batuz