

control/carguero/qr - [generateQrMassive]

PATH: /var/www/html/qaservicesapp/app/Http/Controllers/ShalomControl/CargueroController.php

? Descripción

Genera un registro QR asociado a un **carguero programado (cap_id)**.
Obtiene automáticamente la **placa** del vehículo asignado y registra:

- cap_id
- placa
- usuario
- terminal
- fecha de creación

El registro se almacena en la tabla `emp_cargprogramado_qr` en la base de datos **CentOS**.

Nota: El código contiene una validación de contraseña deshabilitada (comentada).

? Endpoint

POST /generate-qr-massive

? Request Body

```
{
  "password": "string",
  "usuario": "string",
  "cap_id": "number",
  "terminal": "string",
  "placa": "string (ignorado, se reemplaza con la placa obtenida)"
}
```

Reglas

- `password`: opcional actualmente (validación comentada).
- `usuario`: requerido.
- `cap_id`: requerido.
- `terminal`: requerido.
- `placa`: ignorado, se obtiene desde la BD.

? Seguridad

La función no fuerza autenticación ni validación de contraseña porque está comentada, pero originalmente pretendía validar:

```
Hash::check(strtoupper(password), usr_clave)
```

La seguridad dependerá del middleware externo del proyecto.

? Flujo del Servicio (resumen real)

1. Recibe parámetros del request:

`password`, `usuario`, `cap_id`, `terminal`, `placa` (pero placa será ignorada).

2. Obtiene la placa real del carguero desde Empresarial:

```
SELECT cap_cargueroid AS placa
FROM emp_cargprogramado
WHERE cap_id = <cap_id>
```

3. Preparar datos para el QR:

```
{
  cap_id,
  car_placa (placa obtenida),
  user (usuario),
  ter_id (terminal),
  fecha_crea (timestamp actual)
}
```

4. Inserta registro en CentOS:

```
INSERT INTO emp_cargprogramado_qr
RETURNING id
```

- 5. Si la inserción falla → error
Si inserta correctamente → éxito

? Response 200 – Ejemplo

```
{  
  "success": true,  
  "message": "Qr Generado con éxito"  
}
```

? Posibles Errores

1. Error al insertar el registro

```
{  
  "success": false,  
  "message": "Error en el servicio vuelva a intentar"  
}
```

2. cap_id no encontrado (si la consulta no retorna placa)

(El código no valida este error explícitamente, pero sería un caso real posible)

```
{  
  "success": false,  
  "message": "No se encontró carguero asociado al cap_id"  
}
```

3. Error inesperado (excepción del servidor)

(No manejado explícitamente, pero debe considerarse)

```
{  
  "success": false,  
  "message": "Error interno del servidor"  
}
```

? Schemas (estructuras usadas)

emp_cargprogramado

```
{
  "cap_id": "number",
  "cap_cargueroid": "string (placa)"
}
```

emp_cargprogramado_qr

Campos insertados:

```
{
  "cap_id": "number",
  "car_placa": "string",
  "user": "string",
  "ter_id": "string",
  "fecha_crea": "datetime"
}
```

? Lógica en pseudo-código

```
receive password, usuario, cap_id, terminal, placa

// Validar contraseña (actualmente deshabilitado)

placaReal = SELECT placa FROM emp_cargprogramado WHERE cap_id = cap_id

data = {
  cap_id: cap_id,
  car_placa: placaReal,
  user: usuario,
  ter_id: terminal,
  fecha_crea: now()
}

id = INSERT INTO emp_cargprogramado_qr(data)

if !id:
  return error

return éxito
```

Revisión #1

Creado 2025-11-27 11:21:17 -05 por Elian

Actualizado 2025-11-27 11:25:47 -05 por Elian