BY+ · MODULE RESTAURANT · LOT R4

R4 — les API à faire

Le QR de table : le client scanne, voit la carte, appelle le serveur ou demande l'addition. Le serveur le voit. **5 adresses**, plus le travail autour (section 2). Le client ne se connecte pas.

1. Les 5 adresses

1 POST `/restaurant/shops/{shop}/tables/{table}/qr`

Pour le gérant · droit `restaurant.tables.manage` dans le point de vente

Créer ou refaire le QR d'une table.

- Renvoie l'adresse du QR à imprimer, avec le jeton en clair : **une seule fois**.
- On garde en base le jeton **haché** (colonne `restaurant_tables.qr_token`, déjà créée).
- Refaire le QR : l'ancien ne marche plus.

2 GET `/restaurant/public/tables/{token}`

Pour le client · sans connexion · nombre de requêtes limité

La page du QR : la carte du restaurant.

- Renvoie : le nom de la table, le nom et le logo du restaurant, les catégories et leurs plats.
- Pour chaque plat : nom, image, prix du restaurant, accompagnements et suppléments.
- Seulement les articles en vente dans ce restaurant et marqués « en ligne ».
- Jeton inconnu ou refait : **404**.

3 POST `/restaurant/public/tables/{token}/calls`

Pour le client · sans connexion · nombre de requêtes limité

Appeler le serveur ou demander l'addition. Envoi : `{ "kind": "waiter" }` ou `{ "kind": "bill" }`.

- Un seul appel ouvert par table et par type : le client qui appuie 10 fois ne crée qu'un appel.
- L'appel part en direct sur le canal de la salle et les serveurs de la zone sont prévenus (section 2).

4 GET `/restaurant/shops/{shop}/calls`

Pour le serveur · droit `restaurant.orders.take` ou `restaurant.floor.view` dans le point de vente

Les appels en attente, du plus ancien au plus récent.

- Avec `?mine=1` : seulement ceux des zones que sert le serveur connecté.
- Pour chaque appel : la table, le type (serveur ou addition), depuis combien de temps.

5 **POST** `/restaurant/calls/{call}/handle`

Pour le serveur · droit `restaurant.orders.take` dans le point de vente

« Je m'en occupe » : l'appel est pris en charge (qui, quand) et disparaît chez les autres serveurs.

2. Le travail autour (pas de nouvelle adresse)

À faire	Détail
La table `restaurant_calls`	Point de vente, table, commande ouverte de la table (si elle existe), type (`waiter` , `bill`), état (`open` , `handled`), pris en charge par, pris en charge à.
La carte du point de vente **(déjà fournie)**	Pour l'adresse n° 2, utiliser le guichet `Modules\Pos\Public\Contracts\ShopMenu` : `items($shopId, onlineOnly: true)` renvoie les articles et les formules en vente, au prix du point de vente, triés par catégorie puis par nom (`ShopMenuItemData` : type, id, nom, prix, devise, catégorie, image, disponible). Les accompagnements et suppléments : crochet `LineOptions` . **Rien à faire dans le module `pos` .**
Le temps réel	Un événement `restaurant.call.changed` (appel créé, appel pris en charge) sur le canal de la salle, déjà en place : `client.{client}.restaurant.shop.{shop}` . Modèle : `Events/TicketChanged` .
La notification	Aux serveurs de la zone de la table, dans l'application et en push. Modèle : `Services/TicketEvents` .
La pastille	`restaurant.calls_open` (appels en attente dans les zones du serveur), dans `Providers/RestaurantMenuCounter` .
La limite de requêtes	Sur les adresses 2 et 3 (par jeton et par adresse IP). Routes sans connexion : `Route::withoutMiddleware(RequireUser::class)` , comme dans `core/Auth/routes/tenant.php` .

L'appel en cours sur la tuile du plan de salle : fait de notre côté, avec le plan de salle enrichi.

À réutiliser, déjà dans le module : `shopOf()` et `authorizeInShop()` du `Controller` pour les droits ; les services se passent dans les méthodes des contrôleurs, pas dans le constructeur ; une écriture concurrente relit la ligne sous verrou (`lockForUpdate`) ; passer `php artisan tenants:migrate` sur la vraie base.

3. Pas maintenant : V2

> Ne pas faire dans R4 :
>
> - Le compte client : connexion, déconnexion, inscription, suppression (`contact/login` , `contact/logout` …), et toutes les adresses `customer/*` .
> - La commande passée par le client depuis le QR, et sa validation par le serveur.
> - Les horaires d'ouverture, l'addition séparée, les réservations, les statistiques, la reprise de l'ancien projet.