

APPLICATION

- première page de login des serveurs (clic sur le nom)
- Interface avec tous les plats, clic pour ajouter un plat & bouton submit pour envoyer la commande.
- Avant validation, de la commande interface de vérification avec possibilité de supprimer des plats.
- Interface avec toutes les commandes.
- Clic sur une des commandes, accès au récapitulatif.

SERVEUR

- Requête ajax avec API NodeJS
- En cuisine, affichage des commandes, clic sur la commande pour la valider.
- Possibilité de désactiver/réactiver un plat en cliquant dessus.
- Si commande validée, affichage sur nouvelle interface du numéro de la commande/table.
- Stockages des commandes dans une base mongoDB.

2 interfaces :

Backoffice depuis le serveur Node :

Utilisé par le cuisinier pour 2 choses :

- informer les cuisiniers des commandes à réaliser
- informer les serveurs d'une commande prête
- informer les serveurs qu'un plat ne peut plus être réalisé

Application cordova :

Utilisée par les serveur en salle pour :

- identification des serveurs
- envoi de nouvelle commande
- visuel de toute les commandes
- archivage des commandes

Documentation de l'API

GET /api/waiters

/api/waiters

Récupération de tous les serveurs du restaurant.

GET /api/orders

/api/orders

Récupération de toutes les commandes passées par les clients.

GET /api/courses

/api/orders

Récupération de toutes les éléments présents sur le menu.

POST api/orders

api/orders

Ajout d'une commande.

PATCH api/orders

api/orders

Modification d'une commande selon son id.

Passage d'une commande en prête avec l'attribut *prepared* passé à *true*.

Archivage d'une commande payée avec l'attribut *paid* passé à *true*.

PATCH api/courses

api/courses

Modification d'un élément du menu selon son id.

Passage d'un élément en disponible/indisponible avec l'attribut *disponibility* à *true* ou *false*.