



# La gestion zen de vos restaurants

- [Introduction](#)
  - [Introduction de base](#)
  - [Authentification](#)
  - [Limitations](#)
  - [Rejouer les requêtes échouées](#)
- [La carte](#)
  - [GET /catalog/tags](#)
  - [POST /catalog/tags](#)
  - [Entité: Tag](#)
  - [GET /catalog/dishes](#)
  - [POST /catalog/dishes](#)
  - [Entité: Dish](#)
  - [GET /catalog/dishes\\_overrides](#)
  - [POST /catalog/dishes\\_overrides](#)
  - [Entité: DishOverride](#)
  - [GET /catalog/menus](#)
  - [POST /catalog/menus](#)
  - [Entité: Menu](#)
  - [Entité: MenuPart](#)
  - [Entité: Period](#)
  - [Entité: Timetable](#)
  - [GET /catalog/menus\\_overrides](#)
  - [Entité: MenuOverride](#)
  - [GET /catalog/options](#)
  - [POST /catalog/options](#)
  - [Entité: Option](#)
  - [Entité: OptionValue](#)
  - [GET /catalog/options\\_overrides](#)
  - [Entité: OptionOverride](#)
  - [GET /catalog/taxes](#)
  - [GET /catalog/taxes/:id](#)

- [GET /catalogs](#)
- [GET /catalogs/:id](#)
- [Entité: Catalog](#)
- [Les restaurants](#)
  - [GET /restaurants](#)
  - [GET /restaurants/:id](#)
  - [Entité: Restaurant](#)
  - [Entité: Location](#)
  - [Entité: ClosingPeriod](#)
  - [Entité: DeliveryZone](#)
  - [Entité: DeliveryRedirection](#)
  - [GET /restaurants/:id/dishes\\_availability](#)
  - [Entité: DishAvailability](#)
  - [Entité: OptionValueAvailability](#)
  - [GET /fabrication-places](#)
  - [Entité: FabricationPlace](#)
- [Les clients](#)
  - [GET /customers](#)
  - [GET /customers/:id](#)
  - [POST /customers](#)
  - [GET /customers/search](#)
  - [Entité: Customer](#)
  - [Entité: Address](#)
  - [Entité: CustomersSearch](#)
- [Les commandes](#)
  - [GET /orders](#)
  - [GET /orders/:id](#)
  - [POST /orders](#)
  - [PATCH /orders/:id](#)
  - [POST /orders/:id/transactions](#)
  - [POST /orders/:id/closure](#)
  - [POST /orders/:id/production](#)
  - [Champs complémentaires](#)
  - [Entité: OrderEntry \(GET\)](#)
  - [Entité: OrderEntry \(POST\)](#)
  - [Entité: OrderModifier](#)
  - [Entité: OrderEntryMenuPart](#)
  - [Entité: OrderStatus](#)
  - [Entité: Transaction](#)
  - [Entité: Deliveryman](#)
  - [Entité: OrderTransaction](#)
  - [Entité: Tax](#)
- [Les clôtures de caisse](#)
  - [GET /closures](#)
  - [Entité: Closure](#)
- [La réservation](#)
  - [GET /bookings](#)
  - [GET /bookings/:id](#)
  - [POST /bookings](#)
  - [POST /bookings/:id](#)
  - [Entité: Booking](#)
  - [Entité: BookingNew](#)
- [Le plan de salle](#)

- [GET /tableMap/rooms](#)
- [GET /tableMap/rooms/:id](#)
- [Entité: Room](#)
- [Entité: Table](#)
- [Entité: Decoration](#)
- [Les promotions](#)
  - [GET /coupons](#)
  - [GET /coupons/:id](#)
  - [GET /coupon](#)
  - [POST /coupons](#)
  - [POST /canUseCoupon](#)
  - [Entité: Promotion](#)
  - [Entité: RestaurantsValidity](#)
  - [Entité: Item](#)
  - [Entité: promo\\_data](#)
- [Les appels téléphoniques](#)
  - [POST /phonecalls](#)
  - [Entité: Phonecall](#)
- [Les webhooks](#)
  - [Introduction aux Webhooks](#)
  - [Liste des Webhooks](#)
  - [Format des Webhooks](#)
  - [Signature des webhooks](#)
  - [GET /webhooks](#)
  - [POST /webhooks](#)
  - [Entité: Webhook](#)
- [Autres](#)
  - [GET /transaction-methods](#)
  - [GET /transaction-methods/:id](#)
  - [GET /devices](#)
- [Changelog](#)
  - [Version 2.9](#)
  - [Version 2.8](#)
  - [Version 2.7](#)
  - [Version 2.6](#)
  - [Version 2.5](#)
  - [Version 2.4](#)
  - [Version 2.3](#)
  - [Version 2.2](#)
  - [Version 2.1](#)
- [Les erreurs](#)
  - [Entité: Liste des erreurs](#)
  - [Entité: Liste des valeurs errno](#)

---

# Introduction

---

## Introduction de base

URL : <https://api.zelty.fr/<version>/<endpoint>>

**Version actuelle : 2.9 Version actuelle : 2.9**

Toutes les données entrantes ou sortantes sont formatées en JSON et encodées en UTF-8.

Pour chaque appel, est retourné un objet JSON contenant les paramètres suivants :

Nom	Type	Description	Exemple
errno	int	Numéro de l'erreur	5
errmsg	string	La description de l'erreur <i>errno</i>	<i>La clef d'API est incorrecte.</i>

**DÉTAILS SUPPLÉMENTAIRES**

Les attributs non requis, peuvent être omis ou présents avec une valeur nulle.

**VERSIONS DE L'API**

Dans l'URL de chaque endpoint se trouve un numéro de version, qui garanti aux "clients" API qu'une même version continuera de fonctionner de la même manière dans le temps et n'aura pas de "breaking change". Une même version peut en revanche contenir des "non-breaking changes".

La liste des "breaking change" :

- suppression (ou renommage) d'un endpoint
- ajout d'une nouvelle propriété obligatoire
- renommage d'une propriété (en entrée)
- suppression (ou renommage) d'une propriété (en sortie)

Exemple de "non breaking changes" :

- Ajout d'un nouveau endpoint
- Ajout d'une nouvelle propriété facultative (en entrée)
- Ajout d'une nouvelle propriété (en sortie)
- Suppression d'une propriété (si ça n'entraîne pas d'erreur, en sortie)

## Authentification

Chaque appel à l'API doit être authentifié avec le header `Authorization:` .

La valeur de celui-ci peut être récupéré auprès de votre account manager.

**Attention** : La valeur du header est une clef secrète qui doit être utilisée avec précaution (ne pas utiliser dans du Javascript navigateur, ne pas laisser trainer dans du code versionné, etc...).

Voici le guide, sur le support Zelty, permettant au restaurateur de générer une clef API :

<https://support.zelty.fr/hc/fr/articles/360020835771-Comment-g%C3%A9n%C3%A9rer-une-clef-API>

Si vous souhaitez faire des tests de commande, notre équipe technique peut vous créer un accès à un compte test. Pour cela, n'hésitez pas à nous contacter à [tech@zelty.fr](mailto:tech@zelty.fr).

## Limitations

L'API Zety dispose d'une limite quant au nombre de requêtes par seconde.

Actuellement la limite est fixée à 1req/seconde par compte et par IP.

Cette limite est susceptible d'évoluer et peut varier en fonction de la charge des serveurs, particulièrement aux heures de service.

C'est pourquoi il est conseillé de mettre en cache sur vos serveurs des éléments comme le catalogue produit ou la base client.

---

## Rejouer les requêtes échouées

Afin d'anticiper les différents problèmes qui peuvent se produire entre 2 systèmes informatiques : panne réseau, erreur DNS, rate limit, downtime... Il est important de penser à système de *retry* lorsque vous implémentez l'API Zety. Surtout sur les requêtes les plus importantes : par exemple l'envoi d'une nouvelle commande vers Zety.

Nous préconisons d'implémenter un algorithme type *Exponential backoff* et de rejouer chaque requêtes au moins 3 fois en cas d'erreur temporaire.

Le mécanisme de *retry* devrait être implémenté dans les cas suivants :

- En cas de réponse http avec un code d'erreur temporaire : 423, 425, 429, 502, 503
- En cas de serveur joignable : erreur réseau, dns...

Le cas des timeouts et erreurs 5xx :

Seules les requêtes non-idempotentes peuvent être rejouées de manière sécurisée.

Les requêtes non-idempotentes sont :

- Les requêtes avec une des méthodes suivantes : `GET`, `HEAD`, `PUT`, `DELETE`
- Les requêtes POST qui fournissent un `remote_id`. Par exemple lors que vous postez une nouvelle commande (`POST /order s`) avec un paramètre `remote_id` si vous rejoutez la même requêtes Zety vérifiera l'existence de la commande avant de la créer.

Un article intéressant à lire sur le sujet

---

## La carte

---

### GET /catalog/tags

#### Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/tags \
-H 'Authorization: Bearer <token_api>'
```

#### Réponse

```
{
  "tags": [
    {
      "id": 42,
      "id_restaurant": 43,
```

```

    "remote_id": "remote_42",
    "name": "Plats",
    "img": "https://...jpg",
    "description": "Description du tag",
    "id_parent": 483,
    "color": "169c78",
    "o": 0,
    "zc_only": false,
    "zc_hidden": false,
    "meta": null,
  }
}
}

```

Récupération de la liste des tags.

#### PARAMÈTRES GET

Nom	Type	Description	Exemple
show_all	bool	Permet de récupérer tous les tags, inclus ceux qui sont non-visibles sur le web ( <code>"zc_only" : true</code> )	<code>?show_all=1</code>
lang	string	Permet de récupérer certains champs* de la langue indiquée. <div> <div>fr</div> = Français  <div>en</div> = Anglais  <div>es</div> = Espagnol </div>	<code>?lang=fr</code>
all_restaurants	bool	Permet de récupérer les données sur tous les restaurants. <b>Clé API enseigne uniquement</b>	<code>??all_restaurants=1</code>

\*`name` & `description`

#### RÉPONSE

Nom	Type	Description	Exemple
tags	Tag[]	Liste des tags	Voir <b>Entité: Tag</b>

## POST /catalog/tags

#### Exemple de requête

```

curl https://api.zelty.fr/2.9/catalog/tags \
-H 'Authorization: Bearer <token_api>'

```

#### Body

```

{
  "id": 42,
  "remote_id": "remote_42",
  "name": "Plats",
  "img": "https://...jpg",

```

```
{
  "description": "Description du tag",
  "id_parent": 483,
  "color": "169c78",
  "o": 0,
  "zc_only": false,
  "zc_hidden": false
}
```

## Réponse

```
{
  "errno": 0
}
```

Création/modification d'un (ou plusieurs) nouveau(x) tag(s).

### BODY

Le corps de la requête doit contenir un objet de type `Tag` ou une liste d'objets de type `Tag`.  
Les tags disposants d'un `id` ou d'un `remote_id` existants seront mis à jours, les autres créés.

### PARAMÈTRES URL

Nom	Type	Description	Exemple
lang	string	Paramètre optionnel, s'il est renseigné, il remplira certains champs* de la langue indiquée. <code>fr</code> = Français <code>en</code> = Anglais <code>es</code> = Espagnol	<code>?lang=en</code>

\*`name` & `description`

### RÉPONSE

N/A

## Entité: Tag

### Exemple

```
{
  "id": 42,
  "remote_id": "remote_42",
  "name": "Plats",
  "img": "https://....jpg",
  "description": "Description du tag",
  "id_parent": 483,
  "color": "169c78",
  "o": 0,
  "zc_only": false,
  "zc_hidden": false,
  "is_group": false,
  "meta": {...}
}
```

`Tag` représente un groupe ou une catégorie de produits.  
Un produit peut faire parti d'un ou plusieurs tags.

ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int	Yes	ID Zelty	42
remote_id	string		Identifiant distant du tag	remote_42
name	string	Yes	Nom du tag*	Plats
img	string		URL de l'image originale du tag	https://....jpg
description	string		Description du tag*	Description du tag
id_parent	int		ID du tag parent	483
color	string		Couleur en valeur hexadecimal	169c78
o	int		Ordre du tag dans le catalogue	0
zc_only	bool		Disponible uniquement sur la caisse (= non disponible sur le web)	false
zc_hidden	bool		Non-visible sur la caisse	false

\*La valeur peut changer en fonction de la présence du paramètre `lang`.

## GET /catalog/dishes

Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/dishes \
-H 'Authorization: Bearer <token_api>'
```

Réponse

```
{
  "dishes": [
    {
      "id": 42,
      "remote_id": null,
      "id_restaurant": null,
      "sku": null,
      "name": "Produit Zelty",
      "description": "",
      "image": "https://....png",
      "thumb": "https://....png",
      "price": 700,
      "price_togo": null,
      "price_delivery": null,
```



```
    "happy_price": null,
    "cost_price": 200,
    "tax": 1000,
    "tax_takeaway": null,
    "tax_delivery": null,
    "tags": [
      1142,
      1143
    ],
    "options": [
      1042,
      1043
    ],
    "id_fabrication_place": 30,
    "fab_name": null,
    "color": [
      0.9411764705882353,
      0.32941176470588235,
      0.25882352941176473
    ],
    "loyalty_points": 0,
    "loyalty_points_discount": null,
    "earn_loyalty": 0,
    "price_to_define": false,
    "weight_for_price": null,
    "disable": false,
    "disable_takeaway": false,
    "disable_delivery": false,
    "disable_before": null,
    "disable_after": null,
    "o": 0,
    "zc_only": false,
    "meta": {
      "test": "info de test dans meta"
    }
  }
}
```

Récupération de la liste des produits.

PARAMÈTRES GET

Nom	Type	Description	Exemple
show_all	bool	Permet de récupérer tous les produits, inclus ceux qui sont non-visibles sur le web ( <code>"zc_only" : true</code> )	<code>?show_all=1</code>
lang	string	Permet de récupérer certains champs* de la langue indiquée. <code>fr</code> = Français <code>en</code> = Anglais <code>es</code> = Espagnol	<code>?lang=fr</code>
all_restaurants	bool	Permet de récupérer les données sur tous les restaurants. <b>Clé API enseigne uniquement</b>	<code>?all_restaurants=1</code>
limit	int	Limite le nombre de réponse. <i>Par défaut : 2500</i>	<code>?limit=100</code>
offset	int	Permet d'ignorer les premiers résultats en fonction de la valeur renseignée.	<code>?offset=0</code>

\*name & description

RÉPONSE

Nom	Type	Description	Exemple
dishes	Dish[]	Liste des produits	Voir <i>Entité: Dish</i>

## POST /catalog/dishes

Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/dishes \
-H 'Authorization: Bearer <token_api>'
```

Body

```
[
  {
    "id": 42,
    "remote_id": null,
    "id_restaurant": null,
    "sku": null,
    "name": "Produit Zelty",
    "description": "",
    "image": "https://....png",
    "thumb": "https://....png",
    "price": 700,
    "price_togo": null,
    "price_delivery": null,
    "happy_price": null,
    "cost_price": 200,
    "tax": 1000,
    "tax_takeaway": null,
    "tax_delivery": null,
    "tags": [
      1142,
      1143
    ],
    "options": [
      1042,
      1043
    ],
    "id_fabrication_place": 30,
    "fab_name": null,
    "color": [
      0.9411764705882353,
      0.32941176470588235,
      0.25882352941176473
    ],
    "loyalty_points": 0,
    "loyalty_points_discount": null,
    "earn_loyalty": 0,
    "price_to_define": false,
    "weight_for_price": null,
    "disable": false,
  }
]
```

```
[
  {
    "disable_takeaway": false,
    "disable_delivery": false,
    "disable_before": null,
    "disable_after": null,
    "o": 0,
    "zc_only": false,
    "meta": {
      "test": "info de test dans meta"
    }
  }
]
```

## Réponse

```
{
  "errno": 0
}
```

Création/modification d'un (ou plusieurs) produits.

### BODY

Le corps de la requête doit contenir un objet de type `Dish` ou une liste d'objets de type `Dish`.  
Les produits disposants d'un `id` ou d'un `remote_id` existants seront mis à jours, les autres créés.

### PARAMÈTRES URL

Nom	Type	Description	Exemple
lang	string	Paramètre optionnel, s'il est renseigné, il remplira certains champs* de la langue indiquée. <code>fr</code> = Français <code>en</code> = Anglais <code>es</code> = Espagnol	<code>?lang=en</code>

\*`name` & `description`

### RÉPONSE

N/A

## Entité: Dish

### Exemple

```
{
  "id": 42,
  "remote_id": "remote_42",
  "id_restaurant": 43,
  "sku": "01234abcd",
  "name": "Produit Zelty",
  "description": "Ceci est la description du Produit",
  "image": "https://...png",
  "thumb": "https://...png",
  "price": 600,
}
```

```
    "price_togo": 600,
    "price_delivery": 600,
    "happy_price": 900,
    "cost_price": 1100,
    "tax": 1000,
    "tax_takeaway": 1000,
    "tax_delivery": 1000,
    "tags": [
      1142,
      1143
    ],
    "options": [
      1042,
      1043
    ],
    "id_fabrication_place": 30,
    "fab_name": "Cuisine",
    "color": null,
    "loyalty_points": 100,
    "loyalty_points_discount": 500,
    "earn_loyalty": 105,
    "price_to_define": false,
    "weight_for_price": 1000,
    "disable": false,
    "disable_takeaway": false,
    "disable_delivery": false,
    "disable_before": null,
    "disable_after": null,
    "o": -10,
    "zc_only": false,
    "meta": {...}
  }
```

**Dish** représente un produit disponible au catalogue.

ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int		ID Zelty. Lors d'un POST, s'il est présent, cela met à jour un produit existant, sinon cela crée un produit.	42
remote_id	string		Identifiant distant	remote_42
id_restaurant	int		ID du restaurant Si <b>null</b> , le produit est au niveau enseigne.	43
sku	string		Référence code-barre	01234abcd
name	string	Yes	Nom du produit	Produit Zelty
description	string		Description du produit	Ceci est la description du Produit
image	string		URL de l'image originale du produit	https://....png
thumb	string		URL de l'image originale du produit	https://....png

price	int	<b>Yes</b>	Prix TTC du produit <i>En centimes, pas de virgule</i>	600
price_togo	int		Prix TTC du produit pour l'emporter <i>Présent uniquement s'il existe</i>	600
price_delivery	int		Prix TTC du produit pour la livraison <i>Présent uniquement s'il existe</i>	600
happy_price	int		Prix Happy Hour	900
cost_price	int		Coût de revient	1100
tax	int	<b>Yes</b>	Taux de TVA du produit. <i>1000 pour 10%</i>	1000
tax_takeaway	int		Taux de TVA à emporter <i>Présent uniquement s'il existe. 1000 pour 10%</i>	1000
tax_delivery	int		Taux de TVA en livraison <i>Présent uniquement s'il existe. 1000 pour 10%</i>	1000
tags	int[]		Liste des ID des tags du produit	[1142, 1143]
options	int[]		Liste des ID des options du produit <i>20 options max peuvent être associées par produit</i>	[1042, 1043]
id_fabrication_place	int		ID du lieu de fabrication	30
fab_name	string		Nom sur l'écran de cuisine et ticket de production	Cuisine
loyalty_points	int		Points de fidélités nécessaire pour gagner ce produit	100
loyalty_points_discount	int		Remise avec les points de fidélités	500
earn_loyalty	int		Points de fidélités rapportés par ce produit	105
price_to_define	bool		Définir si le Prix est libre. <i>Si le prix est libre, il sera tapé manuellement sur la caisse.</i>	false/true
weight_for_price	int		Poids indiqué par rapport au "Prix TTC". <i>1000 pour 1kg</i>	1000
disable	bool		Disponibilité du produit	false/true
disable_takeaway	bool		Disponibilité du produit sur la vente en ligne à emporter	false/true
disable_delivery	bool		Disponibilité du produit sur la vente en ligne en livraison	false/true
disable_before	datetime		Produit inactif avant cette date <i>Date au format ISO 8601</i>	2017-04-28T19:18:49+00:

			00
disable_after	datetime	Produit inactif après cette date Date au format ISO 8601	2017-04-28T19:18:49+00:00
o	int	Ordre du produit dans le catalogue	-10
zc_only	bool	Disponibilité du produit seulement sur la caisse	false/true
meta	object	Méta données liées au produit Les clefs non renseignées ne seront pas modifiées. Les clefs à supprimer doivent être null.	{"test": "Zelty"}

## GET /catalog/dishes\_overrides

### Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/dishes_overrides \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "dishes_overrides": [
    {
      "dish_id": 42,
      "dish_remote_id": "dish_remote_42",
      "restaurant_id": 43,
      "restaurant_remote_id": "restaurant_remote_43",
      "price": 1120,
      "price_togo": 1120,
      "price_delivery": 1120,
      "tax_id": 742,
      "loyalty_points_discount": 500,
      "disable": false
    }
  ]
}
```

Récupération de la liste des modifications produits par restaurant.

### PARAMÈTRES GET

Nom	Type	Description	Exemple
show_all	bool	Permet de récupérer toutes les surcharges de produits, inclus ceux qui sont non-visibles sur le web ( <code>"zc_only" : true</code> )	?show_all=1

### RÉPONSE

Nom	Type	Description	Exemple
-----	------	-------------	---------

dishes\_override

DishOverride[]

Liste des modifications produits par restaurants

Voir **Entité: DishOverride**

# POST /catalog/dishes\_overrides

## Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/dishes_overrides \
-H 'Authorization: Bearer <token_api>'
```

## Body

```
[
  {
    "restaurant_id": 42,
    "dish_id": 123123,
    "price": 600,
    "disable": false,
    "tax_id": 742,
    "outofstock": false,
    "fabrication_place_name": "Cuisine",
    "id_fabrication_place": 142
  }
]
```

## Réponse

```
{
  "errno": 0
}
```

Modification des infos d'un produit dans un restaurant en particulier.

### BODY

Nom	Type	Requis	Description	Exemple
restaurant_id	int		ID du restaurant concerncé	42
dish_id	int		ID du produit concerné	123123
price	int		Nouveau prix personnalisé à appliquer	600
disable	bool		Disponibilité du produit dans le restaurant	false
tax_id	int		ID du taux de TVA personnalisé à appliquer	742
outofstock	bool		Disponibilité du plat	true
fabrication_place_name	string		Nom du lieu de production <b>Incompatible avec</b> <span>id_fabrication_place</span>	Cuisine

id_fabrication_place	int	ID Zelty du lieu de production	142
Incompatible avec fabrication_place_name			

Pour le champs `price`, la valeur est à renseigner sans décimal  
exemple : 6,00 => `"price": 600`

RÉPONSE

N/A

## Entité: DishOverride

`DishOverride` représente une surcharge de prix d'un produit pour un restaurant.

### Exemple

```
{
  "dish_id": 42,
  "dish_remote_id": "dish_remote_42",
  "restaurant_id": 43,
  "restaurant_remote_id": "restaurant_remote_43",
  "price": 1120,
  "price_togo": 1120,
  "price_delivery": 1120,
  "tax_id": 742,
  "suitepos": 21,
  "loyalty_points_discount": 500,
  "disable": false
}
```

### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
dish_id	int	Yes	ID Zelty	42
dish_remote_id	string		Identifiant distant du produit	dish_remote_42
restaurant_id	int	Yes	ID Zelty Restaurant	43
restaurant_remote_id	string		Identifiant distant du restaurant	restaurant_remote_43
price	int		Prix appliqué par le restaurant. null = prix de l'enseigne	1120
price_togo	int		Prix appliqué par le restaurant. null = prix de l'enseigne	1120
price_delivery	int		Prix appliqué par le restaurant. null = prix de l'enseigne	1120
tax_id	int		ID de la TVA	742
suitepos	int			21



loyalty_points_discount	int	Remise avec les points de fidélités	500
disable	bool	Désactive le produit pour ce restaurant. <i>null = valeur de l'enseigne</i>	false / true / null

## GET /catalog/menus

### Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/menus \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "menus": [
    {
      "id": 42,
      "remote_id": null,
      "name": "Nom du Menu",
      "description": "Description du Menu",
      "image": "https://....png",
      "thumb": "https://....png",
      "price": 600,
      "price_togo": 610,
      "price_delivery": 620,
      "sup": 542,
      "tags": [
        42,
        24
      ],
      "parts": [
        {
          "id": 8461,
          "remote_id": null,
          "name": "Nom de la rubrique menu",
          "description": "Description de la rubrique menu",
          "pos": 0,
          "max": 1,
          "dishes": [
            93877
          ],
          "price": 0,
          "suite": 0,
          "required": false,
          "dishes_prices": {
            "93877": 0
          },
          "position": 10,
          "prevent_customisation": false,
          "has_combinaison": false
        }
      ],
      "enable_hours": {
        "1": [
```

```

    [
      360,
      1439
    ]
  ],
  "2": [
    [
      360,
      1439
    ]
  ],
  "3": [
    [
      360,
      1439
    ]
  ],
  "4": [
    [
      360,
      1439
    ]
  ],
  "5": [
    [
      360,
      1439
    ]
  ],
  "6": [
    [
      0,
      360
    ]
  ],
  "7": []
},
"loyalty_points": 300,
"earn_loyalty": 150,
"price_to_define": false,
"loyalty_points_discount": 500,
"disable": false,
"disable_takeaway": false,
"disable_delivery": false,
"disable_before": null,
"disable_after": null,
"o": 12,
"zc_only": false,
"meta": {}
}
]
```

Récupération de la liste des menus.

PARAMÈTRES GET

Nom	Type	Description	Exemple
show_all	bool	Permet de récupérer tous les menus, inclus ceux qui sont non-visibles sur le web ( <code>"zc_only" : true</code> )	<code>?show_all=1</code>
		Permet de récupérer certains champs* de la langue indiquée.	

lang	string	<div><div>fr</div> = Français</div> <div><div>en</div> = Anglais</div> <div><div>es</div> = Espagnol</div>	?lang=fr
all_restaurants	bool	Permet de récupérer les données sur tous les restaurants. <b>Clé API enseigne uniquement</b>	?all_restaurants=1

\*`name`, `description`, `parts.name` & `parts.description`

RÉPONSE

Nom	Type	Description	Exemple
menus	Menu[]	Liste des menus	Voir <b>Entité: Menu</b>

## POST /catalog/menus

Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/menus \
-H 'Authorization: Bearer <token_api>'
```

Body

```
[
  {
    "id": 42,
    "remote_id": "remote_42",
    "name": "Nom du Menu",
    "description": "Description du Menu",
    "image": "https://....png",
    "thumb": "https://....png",
    "price": 600,
    "price_togo": null,
    "price_delivery": null,
    "sup": 0,
    "tags": [
      42,
      24
    ],
    "parts": [
      {
        "id": 8461,
        "remote_id": null,
        "name": "Nom de la rubrique menu",
        "description": "Description de la rubrique menu",
        "pos": 0,
        "max": 1,
        "dishes": [
          93877
        ],
        "price": 0,
        "suite": 0,
        "required": false,
        "dishes_prices": {
```

```
        "93877": 0
      },
      "position": 10,
      "prevent_customisation": false,
      "has_combinaison": false
    }
  ],
  "enable_hours": {...},
  "loyalty_points": 0,
  "earn_loyalty": 0,
  "price_to_define": false,
  "disable": false,
  "disable_takeaway": false,
  "disable_delivery": false,
  "disable_before": null,
  "disable_after": null,
  "o": 12,
  "meta": {}
}
```

## Réponse

```
{
  "errno": 0
}
```

Création/modification d'un (ou plusieurs) menus.

### BODY

Le corps de la requête doit contenir un objet de type `Menu` ou une liste d'objets de type `Menu`.  
Les menus disposants d'un `id` ou d'un `remote_id` existants seront mis à jours, les autres créés.

### PARAMÈTRES URL

Nom	Type	Description	Exemple
lang	string	Paramètre optionnel, s'il est renseigné, il remplira certains champs* de la langue indiquée. <code>fr</code> = Français <code>en</code> = Anglais <code>es</code> = Espagnol	<code>?lang=en</code>

\*`name` , `description` , `parts.name` & `parts.description`

### RÉPONSE

N/A

## Entité: Menu

### Exemple

```
{
```

```

    "id": 42,
    "remote_id": "remote_42",
    "name": "Nom du Menu",
    "description": "Description du Menu",
    "image": "https://....png",
    "thumb": "https://....png",
    "price": 600,
    "price_togo": null,
    "price_delivery": null,
    "sup": 0,
    "tags": [
      42,
      24
    ],
    "parts": [
      {
        "id": 8461,
        "remote_id": null,
        "name": "Nom de la rubrique menu",
        "description": "Description de la rubrique menu",
        "pos": 0,
        "max": 1,
        "dishes": [
          93877
        ],
        "price": 0,
        "suite": 0,
        "required": false,
        "dishes_prices": {
          "93877": 0
        },
        "position": 10,
        "prevent_customisation": false,
        "has_combinaison": false
      }
    ],
    "enable_hours": {
      "1": [
        [
          360,
          1439
        ]
      ],
      "2": [
        [
          360,
          1439
        ]
      ],
      "3": [
        [
          360,
          1439
        ]
      ],
      "4": [
        [
          360,
          1439
        ]
      ],
      "5": [
        [
          360,
          1439
        ]
      ]
    }
  }

```

```
    ],
    "6": [
      [
        0,
        360
      ]
    ],
    ],
    "7": []
  },
  "loyalty_points": 0,
  "earn_loyalty": 0,
  "price_to_define": false,
  "disable": false,
  "disable_takeaway": false,
  "disable_delivery": false,
  "disable_before": null,
  "disable_after": null,
  "o": 12,
  "meta": {}
}
```

**Menu** représente une formule disponible au catalogue, composée de rubriques et de produits.

ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int	Yes	ID Zelty	42
remote_id	string		Identifiant distant	m43
name	string	Yes	Nom du menu	Nom du Menu
description	string		Description du menu	Description du Menu
image	string		URL de l'image originale du menu	https://....png
thumb	string		URL de l'image en miniature du menu	https://....png
price	int	Yes	Prix TTC du menu. En centimes, pas de virgule	600
price_togo	int		Prix TTC du menu pour l'emporter. Présent uniquement s'il existe	610
price_delivery	int		Prix TTC du menu pour la livraison. Présent uniquement s'il existe	620
sup	int		Supplément (ou remise) à appliquer quand toutes les rubriques <b>has_combinaison</b> sont prises. En centimes, pas de virgule	542
tags	int[]		Liste des ID des tags du menu	[42, 24]
parts	MenuPart[]	Yes	Liste des rubriques qui composent le menu	Voir <b>Entité: MenuPart</b>

enable_hours	Timetable	Créneaux horaire de disponibilité du menu. <i>Vide = actif tout le temps</i>	
loyalty_points	int	Points de fidélités nécessaires pour gagner ce produit	300
earn_loyalty	int	Points de fidélités rapportés par ce menu	150
loyalty_points_discount	int	Remise avec les points de fidélités	500
disable	bool	Disponibilité du menu	false
disable_before	datetime	Produit inactif avant cette date. <i>Date au format ISO 8601</i>	2017-04-28T19:18:49+00:00
disable_after	datetime	Produit inactif après cette date. <i>Date au format ISO 8601</i>	2017-04-28T19:18:49+00:00
o	int	Ordre du produit dans le catalogue	12
zc_only	bool	Disponibilité du menu seulement sur la caisse	false/true
meta	object	Méta données liées au menu <i>Les clefs non renseignées ne seront pas modifiées.</i> <i>Les clefs à supprimer doivent être null.</i>	{"test": "Zelty"}

## Entité: MenuPart

### Exemple

```
"parts": [  
  {  
    "id": 42,  
    "remote_id": "remote_42",  
    "name": "Nom de la rubrique menu",  
    "description": "Description de la rubrique menu",  
    "pos": 0,  
    "max": 1,  
    "dishes": [  
      93877  
    ],  
    "price": 0,  
    "suite": 0,  
    "required": false,  
    "dishes_prices": {  
      "93877": 0  
    },  
    "position": 10,  
    "prevent_customisation": false,  
    "has_combinaison": false  
  }  
]
```

`MenuPart` représente une rubrique dans une formule.

Dans une rubrique, le client peut choisir *X* produit parmi une liste.

ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int	Yes	ID Zelty	42
remote_id	string		ID distant	remote_42
name	string	Yes	Nom de la rubrique	Nom de la rubrique menu
pos	int		Position de la rubrique dans le menu	10
max	int	Yes	Nombre de produit que le client peut choisir dans la rubrique	1
dishes	int[]	Yes	Liste des ID des produits qui constitue la rubrique	[42, 24]
price	int		Supplément si la rubrique est choisie. En centimes, pas de virgule	250
suite	int		Suite de la rubrique (pour la fabrication)	0
required	bool		Cette rubrique est obligatoire	false
dishes_prices	object		Tableau associatif des produits et de leur éventuel supplément. Les clés sont les ID des produits et les valeurs le montant du supplément.	{"42": 250}
has_combinaison	bool		Fait parti des rubriques qui appliquent une remise/supplément	true

Entité: Period

Exemple

```
[
  [
    570,
    900
  ],
  [
    1080,
    1350
  ]
]
```

`Period` représente un créneau horaire.



L'objet `Period` est un tableau de 2 nombres : *horaire de début* et *horaire de fin* (exprimés en minutes).  
Exemple : `[570, 900]` => de 9h30 à 15h

## Entité: Timetable

### Exemple

```
{
  "1": [
    [
      570,
      900
    ],
    [
      1080,
      1350
    ]
  ],
  "2": [
    [
      570,
      900
    ]
  ],
  "3": [
    [
      1080,
      1350
    ]
  ],
  "4": [],
  "5": [],
  "6": [],
  "7": []
}
```

`Timetable` représente une liste de créneaux horaire dans la semaine (du lundi au dimanche).

### ATTRIBUTS

Nom	Type	Description	Exemple
1	Period[]	Liste des horaires pour le lundi <i>Liste des horaires. Chaque horaire est un tableau à 2 entrées</i>	<code>[[570, 900], [1080, 1350]]</code> <i>De 9h30 à 15h et de 18h à 22h30</i>
2	Period[]	Liste des horaires pour le mardi	<i>idem</i>
3	Period[]	Liste des horaires pour le mercredi	<i>idem</i>
4	Period[]	Liste des horaires pour le jeudi	<i>idem</i>
5	Period[]	Liste des horaires pour le vendredi	<i>idem</i>

6	Period[]	Liste des horaires pour le samedi	<i>idem</i>
7	Period[]	Liste des horaires pour le dimanche	<i>idem</i>

## GET /catalog/menus\_overrides

### Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/menus_overrides \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "menus_overrides": [
    {
      "id_menu": 42,
      "menu_remote_id": "remote_42",
      "id_restaurant": 43,
      "restaurant_remote_id": "restaurant_remote_43",
      "price": 1120,
      "price_togo": 1120,
      "price_delivery": 1120,
      "sup": null,
      "loyalty_points_discount": null,
      "disable": true
    }
  ]
}
```

Récupération de la liste des modifications menus par restaurant

#### PARAMÈTRES GET

Nom	Type	Description	Exemple
show_all	bool	Permet de récupérer toutes les surcharges de menus, inclus ceux qui sont non-visibles sur le web ("zc_only" : true)	?show_all=1

#### RÉPONSE

Nom	Type	Description	Exemple
menus_overrides	MenuOverride[]	Liste des modifications menus par restaurants	Voir <b>Entité: MenuOverride</b>

## Entité: MenuOverride

### Exemple

```
{
  "id_menu": 42,
  "menu_remote_id": "remote_42",
  "id_restaurant": 43,
  "restaurant_remote_id": "restaurant_remote_43",
  "price": 1120,
  "price_togo": 1120,
  "price_delivery": 1120,
  "sup": null,
  "loyalty_points_discount": null,
  "disable": true
}
```

`MenuOverride` représente une surcharge de prix d’un menu pour un restaurant.

ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id_menu	int	Yes	ID Zelty du menu	42
menu_remote_id	string		Identifiant distant du menu	remote_42
id_restaurant	int	Yes	ID Zelty Restaurant	43
restaurant_remote_id	string		Identifiant distant du restaurant Zelty	restaurant_remote_43
price	int		Prix appliqué par le restaurant null = prix de l'enseigne	1120
price_togo	int		Prix appliqué par le restaurant null = prix de l'enseigne	1120
price_delivery	int		Prix appliqué par le restaurant null = prix de l'enseigne	1120
sup	int		Supplément (ou remise) à appliquer par le restaurant	null
loyalty_points_discount	int		Remise avec les points de fidélités appliquée par le restaurant	null
disable	bool		Si ce produit est désactivé pour ce restaurant null = valeur de l'enseigne	false / true / null

## GET /catalog/options

Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/options \
-H 'Authorization: Bearer <token_api>'
```

## Réponse

```
{
  "options": [
    {
      "id": 42,
      "remote_id": "remote_42",
      "name": "Sauce",
      "values": [
        {
          "id": 442,
          "remote_id": "Z0V442",
          "name": "Ketchup",
          "description": "Description de l'option Ketchup",
          "image": "",
          "price": 0
        }
      ]
    },
    {
      "nb": 1,
      "mandatory": false,
      "disable": false,
      "zc_only": false,
      "donotpresent": false,
      "meta": {},
      "pick_same_value": true,
      "min_choices": 1,
      "max_choices": 5
    }
  ]
}
```

### Récupération de la liste des options

#### PARAMÈTRES GET

Nom	Type	Description	Exemple
show_all	bool	Permet de récupérer toutes les options, inclus celles qui sont non visibles sur le web (cf. <i>zc_only</i> )	<i>?show_all=1</i>
lang	string	Permet de récupérer certains champs* de la langue indiquée. fr = Français en = Anglais es = Espagnol	<i>?lang=fr</i>
all_restaurants	bool	Permet de récupérer les données sur tous les restaurants. <b>Clé API enseigne uniquement</b>	? <i>all_restaurants=1</i>

\*name, values.name & values.description

#### RÉPONSE

Nom	Type	Description	Exemple
options	Option[]	Liste des options	Voir <b>Entité: Option</b>

# POST /catalog/options

## Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/options \
-H 'Authorization: Bearer <token_api>'
```

## Body

```
[
  {
    "id": 42,
    "remote_id": "Z043",
    "name": "Sauce",
    "values": [
      {
        "id": 442,
        "remote_id": "Z0V43",
        "name": "Ketchup",
        "description": "Description de l'option Ketchup",
        "image": "https://....jpg",
        "price": 0
      }
    ],
    "nb": 1,
    "mandatory": true,
    "disable": false,
    "zcc_only": false,
    "donotpresent": false,
    "meta": {}
  }
]
```

## Réponse

```
{
  "errno": 0
}
```

### Création/modification d'une (ou plusieurs) options

#### BODY

Le corps de la requête doit contenir un objet de type `Option` ou une liste d'objets de type `Option`.  
Les options disposants d'un `id` ou d'un `remote_id` existants seront mise à jours, les autres créées.

#### PARAMÈTRES URL

Nom	Type	Description	Exemple
lang	string	Paramètre optionnel, s'il est renseigné, il remplira certains champs* de la langue indiquée. <code>fr</code> = Français <code>en</code> = Anglais <code>es</code> = Espagnol	<code>?lang=en</code>

\*name , values.name & values.description

RÉPONSE

N/A

# Entité: Option

## Exemple

```
{
  "id": 42,
  "remote_id": "remote_42",
  "name": "Sauce",
  "values": [
    {
      "id": 442,
      "remote_id": "Z0V442",
      "name": "Ketchup",
      "description": "Description de l'option Ketchup",
      "image": "",
      "price": 0
    }
  ],
  "disable": false,
  "zc_only": false,
  "donotpresent": true,
  "meta": {},
  "pick_same_value": true,
  "min_choices": 1,
  "max_choices": 5
}
```

`Option` représente une option dans un produit (choix de la sauce, d'un accompagnement, d'une cuisson...).

ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int	Yes	ID Zelty de l'option	42
remote_id	string		Identifiant distant de l'option	ZO43
name	string	Yes	Nom de l'option	Sauce
values	OptionValue[]		Liste des valeurs possible	Voir <b>Entité: OptionValue</b>
disable	bool		Désactivation de l'option	false
zc_only	bool		Disponible uniquement sur la caisse	false
donotpresent	bool		Ne pas afficher la popup	true
meta	object		Meta données	{"test": "Zelty"}

pick_same_value	int		Autoriser plusieurs fois le même choix si plusieurs réponses sont possibles	<i>true</i>
min_choices	int		Nombre de choix minimum	<i>1</i>
max_choices	int	<b>Yes</b>	Nombre de choix maximum	<i>5</i>

## Entité: OptionValue

### Exemple

```
{
  "id": 442,
  "remote_id": "ZOV442",
  "name": "Ketchup",
  "description": "Description de l'option Ketchup",
  "image": "",
  "price": 0
}
```

`OptionValue` représente une valeur possible dans le choix d'une option.

### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int	<b>Yes</b>	ID Zelty	<i>442</i>
remote_id	string		Identifiant distant	<i>ZOV442</i>
name	string	<b>Yes</b>	Nom de l'option	<i>Ketchup</i>
description	string		Description de l'option	<i>Description de l'option Ketchup</i>
image	string		URL de l'image de l'option	<i>"https://....jpg"</i>
price	int	<b>Yes</b>	Prix de l'option	<i>0</i>

## GET /catalog/options\_overrides

### Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/options_overrides \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "options_overrides": [
    {
      "option_id": 42,
      "restaurant_id": 43,
      "disable": false,
      "min_choices": null,
      "do_not_present": null,
      "zc_only": null
    }
  ]
}
```

Récupération de la liste des modifications options par restaurant

#### PARAMÈTRES GET

Nom	Type	Description	Exemple
show_all	bool	Permet de récupérer toutes les surcharges d'options, inclus ceux qui sont non-visibles sur le web ("zc_only" : true)	?show_all=1

#### RÉPONSE

Nom	Type	Description	Exemple
options_overrides	OptionOverride[]	Liste des modifications options par restaurants	Voir <b>Entité: OptionOverride</b>

## Entité: OptionOverride

### Exemple

```
{
  "option_id": 42,
  "restaurant_id": 43,
  "disable": false,
  "min_choices": null,
  "do_not_present": null,
  "zc_only": null
}
```

`OptionOverride` représente une surcharge d'infos d'une option pour un restaurant.

#### ATTRIBUTS

Nom	Type	Description	Exemple
option_id	int	ID Zelty de l'option	42
restaurant_id	int	ID Zelty Restaurant	43
disable	bool	Si cette option est désactivée pour ce restaurant <i>null = valeur de l'enseigne</i>	false / true / null



min_choices	int	Nombre de choix minimum <i>null = valeur de l'enseigne</i>	<i>false / true / null</i>
do_not_present	bool	Indique si cette option doit être présentée lors de la sélection du produit <i>null = valeur de l'enseigne</i>	<i>false / true / null</i>
zc_only	bool	Disponible uniquement sur la caisse (= non disponible sur le web)	<i>false / true / null</i>

## GET /catalog/taxes

### Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/taxes \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "taxes": [
    {
      "id": 90,
      "name": "TVA 5.50%",
      "rate": 550
    }
  ]
}
```

Récupération de la liste des taxes

### PARAMÈTRES GET

Nom	Type	Description	Exemple
show_all	bool	Permet de récupérer toutes les taxes, inclus celles qui sont désactivées.	<i>?show_all=1</i>
all_restaurants	bool	Permet de récupérer les données sur tous les restaurants. <b>Clé API enseigne uniquement</b>	<i>?all_restaurants=1</i>

### RÉPONSE

Nom	Type	Description	Exemple
taxes	Taxe[]	Liste des taxes	Voir <b>Entité: Tax</b>

## GET /catalog/taxes/:id

### Exemple de requête

```
curl https://api.zelty.fr/2.9/catalog/taxes/<id> \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "tax": {
    "id": 90,
    "name": "TVA 5.50%",
    "rate": 550
  }
}
```

#### Récupération d'une taxe

##### PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int	ID de la Taxe	90

##### PARAMÈTRES GET

N/A

##### RÉPONSE

Nom	Type	Description	Exemple
tax	Taxe	Une taxe	Voir <b>Entité: Tax</b>

## GET /catalogs

### Exemple de requête

```
curl https://api.zelty.fr/2.9/catalogs \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "catalogs": [
    {
      "id": "ab0c00c0-6bcb-4333-b0a0-4d0575237ebd",
      "name": "Catalogue Midi"
    }
  ]
}
```

Récupération de la liste des catalogues.

#### PARAMÈTRES GET

Nom	Type	Description	Exemple
restaurant_id	int	Permet de récupérer tous les catalogues d'un restaurant en particulier. <b>Clé API enseigne uniquement</b>	?restaurant_id=42

#### RÉPONSE

Nom	Type	Description	Exemple
catalogs	Catalog[]	Liste des catalogues de produits	
catalogs.id	string	ID Zelty du Catalogue de produits	ab0c00c0-6bcb-4333-b0a0-4d0575237ebd
catalogs.name	string	Nom du Catalogue de produits	Catalogue Midi

## GET /catalogs/:id

#### Exemple de requête

```
curl https://api.zelty.fr/2.9/catalogs/<id> \
-H 'Authorization: Bearer <token_api>'
```

#### Réponse

```
{
  "catalog": {
    "id": "ab0c00c0-6bcb-4333-b0a0-4d0575237ebd",
    "brand_id": 2,
    "restaurant_id": 42,
    "locale": "fr_FR",
    "name": "Catalogue Midi",
    "description": "Description du catalogue",
    "currency": "EUR",
    "tags": [{...}],
    "items": [{...}],
    "optionValues": [{...}],
    "options": [{...}],
    "menuParts": [{...}],
    "taxes": [{...}]
  },
  "errno": 0
}
```

Récupération d'un catalogue.

#### PARAMÈTRES URL

Nom	Type	Description	Exemple

id	string	ID du catalogue	ab0c00c0-6bcb-4333-b0a0-4d0575237ebd
----	--------	-----------------	--------------------------------------

RÉPONSE

Nom	Type	Description	Exemple
catalog	Catalog	Détails d'un catalogue de produits	Voir <i>Entité: Catalog</i>

# Entité: Catalog

Exemple

```
{
  "id": "ab0c00c0-6bcb-4333-b0a0-4d0575237ebd",
  "brand_id": 2,
  "restaurant_id": 42,
  "locale": "fr_FR",
  "name": "Catalogue Midi",
  "description": "Description du catalogue",
  "currency": "EUR",
  "tags": [
    {
      "internal_id": "141205",
      "id": "ZT141205",
      "external_id": null,
      "parent_id": null,
      "name": "Catégorie 1",
      "description": "Description de cat 1",
      "image": null,
      "color": "#ff00ff",
      "item_ids": [
        "ZD1989",
        "ZD1976",
        "ZD1978"
      ],
      "disabled": false,
      "metadata": {}
    }
  ],
  "items": [
    {
      "internal_id": "1989",
      "id": "ZD1989",
      "type": "dish",
      "external_id": null,
      "name": "Soupe Miso",
      "pos_name": null,
      "kitchen_name": "",
      "description": null,
      "img": "uploads/images/34/00c1.png",
      "color": null,
      "disabled": true,
      "is_sold_out": false,
      "price": {
        "price": 300,
        "is_fixed": true,
        "prevent_discounts": false,
        "overrides": []
      }
    }
  ]
}
```

```

    },
    "availability": {
      "start_date": null,
      "end_date": null
    },
    "loyalty": null,
    "metadata": {},
    "tax_rules": {
      "tax_id": "ZTX703",
      "overrides": []
    }
  }
],
"optionValues": [
  {
    "internal_id": "32933",
    "id": "ZOV32933",
    "type": "option_value",
    "external_id": null,
    "name": "Sans sauce",
    "pos_name": null,
    "kitchen_name": null,
    "description": null,
    "img": null,
    "color": null,
    "disabled": false,
    "is_sold_out": false,
    "price": {
      "price": 0,
      "is_fixed": true,
      "prevent_discounts": false,
      "overrides": []
    },
    "availability": null,
    "loyalty": null,
    "metadata": {}
  }
],
"options": [
  {
    "internal_id": "5517",
    "id": "Z05517",
    "type": "option",
    "external_id": null,
    "name": "Sans",
    "minimum_choices": 0,
    "maximum_choices": 3,
    "pick_same_value": true,
    "disabled": false,
    "value_ids": [
      "ZOV32933",
      "ZOV32934",
      "ZOV32935"
    ],
    "pos_settings": true,
    "metadata": {}
  }
],
"menuParts": [
  {
    "internal_id": "44869",
    "id": "ZMP44869",
    "type": "menu_part",
    "external_id": null,
    "name": "Rubri UE 1 (5 max - obli)",

```

```
    "pos_name": null,
    "kitchen_name": null,
    "description": null,
    "img": null,
    "color": null,
    "disabled": false,
    "is_sold_out": false,
    "price": {
      "price": 0,
      "is_fixed": true,
      "prevent_discounts": false,
      "overrides": []
    },
    "availability": null,
    "loyalty": null,
    "metadata": {},
    "minimum_choices": 5,
    "maximum_choices": 5,
    "dish_ids": [
      "ZD694164",
      "ZD694166",
      "ZD694168"
    ]
  },
],
"taxes": [
  {
    "id": "ZTX703",
    "name": "TVA 10.00%",
    "rate": 1000
  }
]
}
```

`Catalog` représente un catalogue de carte. Les catalogues ne sont modifiables que sur le BO Zelty.

ATTRIBUTS

Nom	Type	Description	Exemple
id	string	ID Zelty du Catalogue de produits	<i>ab0c00c0-6bcb-4333-b0a0-4d0575237ebd</i>
brand_id	int	ID Zelty de l'enseigne	<i>2</i>
restaurant_id	int	ID Zelty du restaurant	<i>42</i>
locale	string	Langue du catalogue	<i>fr_FR</i>
name	string	Nom du Catalogue	<i>Catalogue Midi</i>
description	string	Description du catalogue de produits	<i>Description du catalogue</i>

# Les restaurants

## GET /restaurants

Exemple de requête

```
curl https://api.zelty.fr/2.9/restaurants \
-H 'Authorization: Bearer <token_api>'
```

Réponse

```
{
  "restaurants": [
    {
      "id": 42,
      "remote_id": "remote_42",
      "disable": false,
      "name": "Zelty Resto",
      "description": "Description du restaurant",
      "country_code": "FR",
      "currency": "EUR",
      "image": "https://....png",
      "default_lang": "fr",
      "takeaway_delay": 15,
      "delivery_time": 45,
      "delay": 15,
      "delivery_charge": 0,
      "delivery_charge_tva": 2000,
      "delivery_minimum": 1500,
      "delivery_no_charge_min": 2500,
      "delivery_hours": { ... },
      "opening_hours": { ... },
      "opening_hours_txt": "Du lundi au samedi de 9h à 15h",
      "happy_hours": { ... },
      "closures": [],
      "address": "61 Rue de Cléry 75002 Paris",
      "phone": "+33123456789",
      "public_name": "Restaurant Zelty",
      "online_ordering_hidden": false,
      "loc": { ... }
    }
  ]
}
```

Récupération des restaurants

PARAMÈTRES GET

Nom	Type	Description	Exemple
includeRedirections	bool	Rajoute la liste des redirections à la réponse. <b>Clé API enseigne uniquement</b>	<i>includeRedirections=true</i>
fields[]	string[]	Rajoute la liste des zones de livraisons	<i>fields[]=delivery_zones</i>
fields[]	string[]	Rajoute les informations de facturation	<i>fields[]=billing</i>

RÉPONSE

Nom	Type	Description	Exemple
restaurants	Restaurant[]	Liste des restaurants du groupe	Voir <b>Entité: Restaurant</b>

# GET /restaurants/:id

## Exemple de requête

```
curl https://api.zelty.fr/2.9/restaurants/<id> \
-H 'Authorization: Bearer <token_api>'
```

## Réponse

```
{
  "restaurant": {
    "id": 42,
    "remote_id": "remote_42",
    "disable": false,
    "name": "Zelty Resto",
    "description": "Description du restaurant",
    "country_code": "FR",
    "currency": "EUR",
    "image": "https://....png",
    "default_lang": "fr",
    "takeaway_delay": 15,
    "delivery_time": 45,
    "delay": 15,
    "delivery_charge": 0,
    "delivery_charge_tva": 2000,
    "delivery_minimum": 1500,
    "delivery_no_charge_min": 2500,
    "delivery_hours": { ... },
    "opening_hours": { ... },
    "opening_hours_txt": "Du lundi au samedi de 9h à 15h",
    "happy_hours": { ... },
    "closures": [],
    "address": "61 Rue de Cléry 75002 Paris",
    "phone": "+33123456789",
    "public_name": "Restaurant Zelty",
    "online_ordering_hidden": false,
    "loc": { ... }
  }
}
```

### Récupération des détails d'un restaurant

#### PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int / string	<code>id</code> (int) ou <code>remote_id</code> (string) du restaurant	42

#### PARAMÈTRES GET

Nom	Type	Description	Exemple
fields[]	string[]	Rajoute la liste des zones de livraisons	<code>fields[]=delivery_zones</code>
fields[]	string[]	Rajoute les informations de facturation	<code>fields[]=billing</code>



RÉPONSE

Nom	Type	Description	Exemple
restaurant	Restaurant	Détails d'un restaurant	Voir <i>Entité: Restaurant</i>

Entité: Restaurant

Exemple

```
{
  "id": 42,
  "remote_id": "remote_42",
  "disable": false,
  "name": "Zelty Resto",
  "description": "Description du restaurant",
  "country_code": "FR",
  "currency": "EUR",
  "image": "https://....png",
  "default_lang": "fr",
  "takeaway_delay": 15,
  "delivery_time": 45,
  "delay": 15,
  "delivery_charge": 0,
  "delivery_charge_tva": 2000,
  "delivery_minimum": 1500,
  "delivery_no_charge_min": 2500,
  "delivery_hours": { ... },
  "opening_hours": { ... },
  "opening_hours_txt": "Du lundi au samedi de 9h à 15h",
  "happy_hours": { ... },
  "closures": [],
  "address": "61 Rue de Cléry 75002 Paris",
  "phone": "+33123456789",
  "public_name": "Restaurant Zelty",
  "online_ordering_hidden": false,
  "loc": { ... },
  "delivery_zones": [
    { ... }
  ]
}
```

**Restaurant** représente un restaurant (contrairement à ce que le nom pourrait laisser penser).

ATTRIBUTS

Nom	Type	Description	Exemple
id	int	ID Zelty	42
remote_id	string	Identifiant distant	remote_42
disable	bool	Magasin actif ou non	false
name	string	Nom du restaurant	Zelty Resto

description	string	Description du restaurant	<i>Description du restaurant</i>
country_code	string	Pays du restaurant au format ISO	<i>FR</i>
currency	string	Monnaie du restaurant au format ISO	<i>EUR</i>
image	string	URL de l'image du restaurant	<i>https://....png</i>
default_lang	string	Langue du restaurant	<i>fr</i>
takeaway_delay	int	Délai pour la commande à emporter (en minutes)	<i>15</i>
delivery_time	int	Délai par défaut pour la livraison	<i>45</i>
delay	int	Délai de production supplémentaire (temporaire)	<i>15</i>
ordering_available	bool	Activer ou non la Commande en ligne Zelty	<i>true</i>
ordering_delay	int	Indique s'il y a ou non un Délai supplémentaire pour la Commande en ligne	<i>5</i>
delivery_charge	int	Frais de livraison	<i>0</i>
delivery_charge_tva	int	TVA pour les frais de livraison	<i>2000</i>
delivery_minimum	int	Montant minimum pour la livraison	<i>1500</i>
delivery_no_charge_minimum	int	Montant à partir duquel les frais de livraison sont offerts	<i>2500</i>
delivery_hours	Timetable	Créneaux horaires de livraison	<i>Voir <b>Entité: Timetable</b></i>
opening_hours	Timetable	Horaires d'ouverture (pour la vente à emporter)	<i>Voir <b>Entité: Timetable</b></i>
opening_hours_txt	string	Horaires d'ouverture textuels	<i>Du lundi au samedi de 9h à 15h</i>
happy_hours	Timetable	Horaires du Happy Hours	<i>Voir <b>Entité: Timetable</b></i>
closures	ClosingPeriod[]	Liste des fermetures exceptionnelles	<i>Voir <b>Entité: ClosingPeriod</b></i>
address	string	Adresse du restaurant	<i>61 Rue de Cléry 75002 Paris</i>
phone	string	Numéro de téléphone du restaurant	<i>+33123456789</i>
public_name	string	Nom public du restaurant	<i>Restaurant Zelty</i>
online_ordering_hidden	bool	Masquer le restaurant de la commande en ligne	<i>false</i>

loc	Location	Coordonnées GPS	Voir <b>Entité: Location</b>
delivery_zones	DeliveryZone[]	Liste des zones de livraison (Read Only)	Voir <b>Entité: DeliveryZone</b>

## Entité: Location

### Exemple

```
{
  "lat": 48.8923147,
  "lng": 2.354988
}
```

**Location** représente des coordonnées GPS.

#### ATTRIBUTS

Nom	Type	Description	Exemple
lat	Float	Latitude	48.8923147
lng	Float	Longitude	2.354988

## Entité: ClosingPeriod

### Exemple

```
{
  "start": 1497953059,
  "end": 1497953059,
  "reason": "Fermé pour travaux"
}
```

**ClosingPeriod** représente une fermeture exceptionnelle.

#### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
start	int	<b>Yes</b>	Timestamp du début de la fermeture (en secondes, UTC)	1497953059
end	int	<b>Yes</b>	Timestamp de la fin de la fermeture (en secondes, UTC)	1497953059
reason	string		Message informatif concernant la fermeture	Fermé pour travaux

# Entité: DeliveryZone

## Exemple

```
{
  "id": 42,
  "id_restaurant": 43,
  "name": "18ème",
  "type": 1,
  "zip_code": "75018",
  "city": "Paris",
  "center": null,
  "polygon": [
    {
      "lat": 48.8327872,
      "lng": 2.2863048
    }
  ],
  "allstreets": true,
  "delivery_time": 45,
  "delivery_charge": 0,
  "force_web_payment": 0,
  "min_price": 15,
  "closures": null,
  "disabled": false
}
```

`DeliveryZone` représente une zone de livraison d'un restaurant.

### ATTRIBUTS

Nom	Type	Description	Exemple
id	int	ID Zelty	42
id_restaurant	int	ID Zelty Restaurant	43
name	string	Nom de la zone	18ème
type	int	Type de zone 0 = Zone Code postal/Ville 1 = Zone Google Maps	1
zip_code	string	Code postal de la zone. Uniquement si "type": 0	75018
city	string	Ville de la zone. Uniquement si "type": 0	Paris
polygon	Polygon	Polygone du tracé de la zone	Voir <b>Entité: Location</b>
delivery_time	int	Délai de livraison (en minutes)	45
delivery_charge	int	Frais de livraison	0
force_web_payment	bool	Paiement en ligne forcé sur cette zone	false

min_price	int	Minimum de commande pour cette zone	15
closures	Closure[]	Liste des fermetures exceptionnelles	Voir <b>Entité: ClosingPeriod</b>
disabled	bool	Zone inactive	false

## Entité: DeliveryRedirection

### Exemple

```
{
  "id": 42,
  "name": "Resto 1 vers Resto 2",
  "id_restaurant": 43,
  "opening_hours": {
    "1": [
      [
        682,
        1413
      ]
    ],
    "2": [],
    "3": [],
    "4": [],
    "5": [],
    "6": [],
    "7": []
  },
  "id_zones": [
    42,
    24
  ],
  "disable": false
}
```

**DeliveryRedirection** représente une redirection de zones vers un autre restaurant.

### ATTRIBUTS

Nom	Type	Description	Exemple
id	int	ID Zelty	42
name	string	Nom de la redirection	Resto 1 vers Resto 2
id_restaurant	int	ID Zelty Restaurant vers lequel les zones sont redirigées	43
opening_hours	Timetable	Horaires d’activation de la redirection	Voir <b>Entité: Timetable</b>
id_zones	int[]	Liste des ID des zones concernées	[42, 24]
disable	bool	Redirection inactive	false

# GET /restaurants/:id/dishes\_availability

## Exemple de requête

```
curl https://api.zelty.fr/2.9/restaurants/<id>/dishes_availability \
-H 'Authorization: Bearer <token_api>'
```

## Réponse

```
{
  "dishes": [
    {
      "id_dish": 42,
      "dish_remote_id": "dish_remote_42",
      "id_restaurant": 43,
      "restaurant_remote_id": "restaurant_remote_43",
      "outofstock": false
    }
  ],
  "options_values": [
    {
      "id_option": 42,
      "option_remote_id": "option_remote_42",
      "id_option_value": 44,
      "option_value_remote_id": "option_value_remote_44",
      "outofstock": false,
      "id_restaurant": 43,
      "restaurant_remote_id": "restaurant_remote_43"
    }
  ]
}
```

Récupération de la liste des produits actifs/inactifs par restaurant

### PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int	ID Zelty du restaurant	42

### PARAMÈTRES GET

N/A

### RÉPONSE

Nom	Type	Description	Exemple
dishes	DishAvailability[]	Liste des dispo des produits par restaurant.	Voir <b>Entité: DishAvailability</b>
options_values	OptionValueAvailability[]	Liste des dispo des valeurs des options par restaurant.	Voir <b>Entité: OptionValueAvailability</b>

## Entité: DishAvailability

### Exemple

```
{
  "id_dish": 42,
  "dish_remote_id": "dish_remote_42",
  "id_restaurant": 43,
  "restaurant_remote_id": "restaurant_remote_43",
  "outofstock": false
}
```

*DishAvailability* représente la disponibilité d'un produit pour un restaurant.

### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id_dish	int	<b>Yes</b>	ID Zelty	42
dish_remote_id	string		Identifiant distant du produit	<i>dish_remote_42</i>
id_restaurant	int	<b>Yes</b>	ID Zelty Restaurant	43
restaurant_remote_id	string		Identifiant distant du restaurant	<i>restaurant_remote_43</i>
outofstock	bool	<b>Yes</b>	Si le produit est indisponible outofstock = true	<i>false</i>

## Entité: OptionValueAvailability

### Exemple

```
{
  "id_option": 42,
  "option_remote_id": "option_remote_42",
  "id_option_value": 44,
  "option_value_remote_id": "option_value_remote_44",
  "outofstock": false,
  "id_restaurant": 43,
  "restaurant_remote_id": "restaurant_remote_43"
}
```

*OptionValueAvailability* représente la disponibilité d'une valeur d'une option pour un restaurant.

### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id_option	int	<b>Yes</b>	ID Zelty de l'option	42

option_remote_id	string		Identifiant distant de la valeur	<i>option_remote_42</i>
id_option_value	int	<b>Yes</b>	ID Zelty de la valeur de l'option	<i>44</i>
option_value_remote_id	string		Identifiant distant de la valeur	<i>option_value_remote_44</i>
outofstock	bool	<b>Yes</b>	Indique si le produit est indisponible	<i>false</i>
id_restaurant	int	<b>Yes</b>	ID Zelty Restaurant	<i>43</i>
restaurant_remote_id	string		Identifiant distant du restaurant	<i>restaurant_remote_43</i>

## GET /fabrication-places

### Exemple de requête

```
curl https://api.zelty.fr/2.9/fabrication-places \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "data": [
    {
      "id": 42,
      "name": "Cuisine"
    }
  ]
}
```

Récupération des lieux de fabrication de produit

#### PARAMÈTRES GET

N/A

#### RÉPONSE

Nom	Type	Description	Exemple
data	FabricationPlace[]	Liste des lieux de fabrication	Voir <b>Entité: FabricationPlace</b>

## Entité: FabricationPlace

### Exemple

```
{
```



```
{
  "id": 42,
  "name": "Cuisine"
}
```

**FabricationPlace** représente un lieu de fabrication de produit.

#### ATTRIBUTS

Nom	Type	Description	Exemple
id	int	ID Zelty du lieu de fabrication	42
name	string	Nom du lieu de fabrication	Cuisine

## Les clients

### GET /customers

#### Exemple de requête

```
curl https://api.zelty.fr/2.9/customers \
-H 'Authorization: Bearer <token_api>'
```

#### Réponse

```
{
  "customers": [
    {
      "id": 123456,
      "uuid": "abcdef-1234-1234-azertyuiop",
      "remote_id": "remote_123456",
      "nice_name": "Toto Test",
      "updated_at": "2020-01-02T12:57:23+0100",
      "name": "Test",
      "fname": "Toto",
      "company": "Zelty",
      "card": "abcd98765",
      "phone": "+33612345678",
      "phone2": "+33600000002",
      "mail": "toto.test@zelty.fr",
      "birthday": "2000-12-31",
      "balance": 1500,
      "personal_info": "Allergique aux légumes",
      "loyalty": 420,
      "registration": "2019-06-14T18:51:35+02:00",
      "default_address": 0,
      "billing_address": 0,
      "sms_optin": null,
      "mail_optin": null,
      "turnover": 4200,
      "nb_orders": 3,
      "last_order_date": "2019-06-17T14:28:21+02:00",
      "vip": false,
    }
  ]
}
```

```
    "other": "",
    "metadata": {
      "meta0": 1234,
      "meta1": "champs meta 1",
      "meta2": "champs meta 2"
    },
    "addresses": [
      { ... }
    ]
  }
}
```

Récupération de la liste des clients

PARAMÈTRES GET

Paramètre	Description	Exemple
cardnum	Filtrer par numéro de carte de fidélité. customer.card	?cardnum=abcd98765
mail	Filtrer par email. customer.mail	?mail=toto.test@zelty.fr
phone	Filtrer par numéro de téléphone. customer.phone uniquement	?phone=+33612345678
birthday	Filtrer par date de naissance. Format YYYY-mm-dd ou mm-dd	?birthday=2000-12-31
search	Rechercher par nom/prénom/entreprise	?search=toto
limit	Limite le nombre de réponse. Par défaut : 100, Max : 1000	?limit=50
offset	Permet d'ignorer les premiers résultats en fonction de la valeur renseignée.	?offset=0
updated_before	Filtrer uniquement les fiches créées/modifiées avant cette date : YYYY-mm-dd Format ISO 8601	?updated_before=
updated_after	Filtrer uniquement les fiches créées/modifiées après cette date : YYYY-mm-dd Format ISO 8601	?updated_after=

RÉPONSE

Nom	Type	Description	Exemple
customer	Customer[]	Fiche client	Voir <b>Entité: Customer</b>

# GET /customers/:id

## Exemple de requête

```
curl https://api.zelty.fr/2.9/customers/<id> \
-H 'Authorization: Bearer <token_api>'
```

## Réponse

```
{
  "customer": {
    "id": 123456,
    "uuid": "abcdef-1234-1234-azertyuiop",
    "remote_id": "remote_123456",
    "nice_name": "Toto Test",
    "updated_at": "2020-01-02T12:57:23+0100",
    "name": "Test",
    "fname": "Toto",
    "company": "Zelty",
    "card": "abcd98765",
    "phone": "+33612345678",
    "phone2": "+336000000002",
    "mail": "toto.test@zelty.fr",
    "birthday": "2000-12-31",
    "balance": 1500,
    "personal_info": "Allergique aux légumes",
    "loyalty": 420,
    "registration": "2019-06-14T18:51:35+02:00",
    "default_address": 0,
    "billing_address": 0,
    "sms_optin": null,
    "mail_optin": null,
    "turnover": 4200,
    "nb_orders": 3,
    "last_order_date": "2019-06-17T14:28:21+02:00",
    "vip": false,
    "other": "",
    "metadata": {
      "meta0": 1234,
      "meta1": "champs meta 1",
      "meta2": "champs meta 2"
    },
    "addresses": [
      { ... }
    ]
  }
}
```

Récupération d'une fiche client

### PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int / string	ID (int) ou remote_id (string) du client	123456 / remote_123456

### PARAMÈTRES GET

N/A

## RÉPONSE

Nom	Type	Description	Exemple
customer	Customer	Fiche client	Voir <b>Entité: Customer</b>

## POST /customers

### Exemple de requête

```
curl https://api.zelty.fr/2.9/customers \
-H 'Authorization: Bearer <token_api>'
```

### Body

```
{
  "id": 123456,
  "name": "Test",
  "fname": "Toto",
  "company": "Zelty",
  "metadata": {
    "meta0" : 1234,
    "meta1" : "champs meta 1",
    "meta2" : "champs meta 2"
  },
  "addresses": [
    { ... }
  ]
}
```

### Réponse

```
{
  "errno": 0
}
```

### Création/modification d'une fiche client

#### BODY

Le corps de la requête doit contenir un objet de type `Customer`.  
Une seule requête par client.

#### PARAMÈTRES URL

Nom	Type	Description	Exemple
check_existing	bool	Permet de vérifier l'existence de la fiche client : - via son email ( <code>mail</code> ) dans un premier temps, puis - via son n° de téléphone ( <code>phone</code> ) dans un second temps.	<code>check_existing=1</code>

Si le client existe : la requête va mettre à jour la fiche client, sinon la requête va créer une nouvelle fiche client.

RÉPONSE

N/A

## GET /customers/search

Exemple de requête

```
curl https://api.zelty.fr/2.9/customers/search?term=toto \
-H 'Authorization: Bearer <token_api>'
```

Réponse

```
{
  "customers": [
    {
      "id": 123456,
      "uuid": "...",
      "remote_id": "remote_123456",
      "nice_name": "Toto Test",
      "card": "abcd98765",
      "phone": "+33612345678",
      "mail": "toto.test@zelty.fr"
    }
  ]
}
```

Permet de chercher un client en fonction du filtre utilisé

PARAMÈTRES GET

Paramètre	Description	Exemple
cardnum	Filtrer par numéro de carte de fidélité. customer.card	?cardnum=abcd98765
mail	Filtrer par email. customer.mail	?mail=toto.test@zelty.fr
term	Rechercher par noms/prénoms/entreprise/mail	?term=<name/fname/company/mail>
limit	Limite le nombre de réponse. Par défaut : 15, Max : 50	?limit=20

RÉPONSE

Nom	Type	Description	Exemple
customer	CustomersSearch[]	Fiche client	Voir <b>Entité: CustomersSearch</b>

# Entité: Customer

`Customer` représente un client.

## Exemple

```
{
  "id": 123456,
  "uuid": "abcdef-1234-1234-1234-azertyuiop",
  "remote_id": "remote_123456",
  "nice_name": "Toto Test",
  "updated_at": "2020-01-02T12:57:23+0100",
  "name": "Test",
  "fname": "Toto",
  "company": "Zelty",
  "card": "abcd98765",
  "phone": "+33612345678",
  "phone2": "+33600000002",
  "mail": "toto.test@zelty.fr",
  "birthday": "2000-12-31",
  "balance": 1500,
  "personal_info": "Allergique aux légumes",
  "loyalty": 420,
  "registration": "2019-06-14T18:51:35+02:00",
  "default_address": 0,
  "billing_address": 0,
  "sms_optin": null,
  "mail_optin": null,
  "turnover": 4200,
  "nb_orders": 3,
  "last_order_date": "2019-06-17T14:28:21+02:00",
  "vip": false,
  "other": "",
  "metadata": {
    "meta0": 1234,
    "meta1": "champs meta 1",
    "meta2": "champs meta 2"
  },
  "addresses": [
    { ... }
  ]
}
```

## ATTRIBUTES

Nom	Type	Requis	Description	Exemple
id	int	Yes	ID Zelty (read-only) Si le champs est présent, cela mettre à jour une fiche client, s'il est absent, cela créera une nouvelle fiche.	123456
remote_id	string		Identifiant distant	remote_123456
nice_name	string		Prénom & Nom	Toto Test
updated_at	datetime		Date de la dernière mise à jour (read-only) Date au format ISO 8601	2020-01-02T12:57:23+0100

name	string	Yes	Nom Seule une de ces 3 informations est obligatoire : <input type="text"/> <input type="text"/> <input type="text"/> e, <input type="text"/> & <input type="text"/>	Test
fname	string	Yes	Prénom Seule une de ces 3 informations est obligatoire : <input type="text"/> <input type="text"/> <input type="text"/> e, <input type="text"/> & <input type="text"/>	Toto
company	string	Yes	Nom de société Seule une de ces 3 informations est obligatoire : <input type="text"/> <input type="text"/> <input type="text"/> e, <input type="text"/> & <input type="text"/>	Zelty
card	string		Numéro de carte de fidélité. Max : 128 caractères. Ce champs peut être édité lorsqu'il est vide. S'il contient déjà une valeur, ce champs ne pourra ni être modifié, ni être supprimé via l'API. Il pourra être supprimé uniquement manuellement sur le BO Zelty.	abcd98765
phone	string		Numéro de téléphone. Format international E164	+33612345678
phone2	string		Numéro de téléphone. Format international E164	+33600000002
mail	string		Email du client	toto.test@zelty.fr
birthday	date		Date de naissance du client. Format <input type="text"/>	2000-12-31
balance	int		Solde du client en centimes	1500
personnal_info	date		Commentaire client	Allergique aux légumes
loyalty	int		Solde de points de fidélités (read-only)	420
registration	datetime		Date de création de la fiche client (read-only) Date au format ISO 8601	2019-06-14T18:51:35+02:00
default_address	int		Adresse par défaut (ID de l'adresse à renseigner)	0
billing_address	int		Adresse de facturation par défaut (ID de l'adresse à renseigner)	0
sms_optin	bool		Opt-in SMS	false/true/null
mail_optin	bool		Opt-in Mail	false/true/null
turnover	int		Chiffre d'affaires (en centimes) cumulé généré par le client (read-only)	4200
nb_orders	int		Nombre total de commandes effectuées par le client (read-only)	3
			Date de la dernière commande (read-only)	2017-04-

last_order_date	datetime	Date au format ISO 8601	28T19:18:49+00:00
other	string	Information interne, qui apparaîtra en pop-up lors de la prochaine selection du client sur la caisse	Ce client possèdent un avoir
metadata	list	Liste de (15, au maximum) <Clé> : <Valeur>. <Clé> peut avoir 40 caractères max. <Valeur> peut avoir 200 caractères max. Le champs metadata peut-être utilisé pour stocker une donnée complémentaire sur l'objet Customer	{ "order_id" : "1234", "intern" : "sample", "status" : "Novice" }
last_restaurant_id	int	ID du dernier restaurant fréquenté par le client	
addresses	Address[]	Liste des adresses du client	Voir <b>Entité: Address</b>
country_code	date	Code pays ISO du client	FR

## Entité: Address

### Exemple

```
{
  "id": 42,
  "remote_id": "m43",
  "name": "Maison",
  "street": "Rue de Cléry",
  "street_num": "61",
  "zip_code": "75002",
  "city": "Paris",
  "formatted_address": "61 Rue de Cléry, 75002 Paris, France",
  "google_id": "ChIJQTUM5BZu5kcR4Z7u0kT8CMk",
  "location": [
    48.8689952,
    2.349442800000002
  ],
  "address_more": "Rue de Cléry",
  "floor": "Rez-de-chaussée",
  "door": "101",
  "building": "A",
  "code": "75A"
}
```

**Address** représente une adresse client.

### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int	Yes	ID Zelty	42
remote_id	string		Identifiant distant	m43



name	string	Yes	Nom de l'adresse	Maison
street	string	Yes	Nom de la voie	Rue de Cléry
street_num	string		Numéro de l'adresse dans la voie	61
zip_code	string	Yes	Code postal	75002
city	string	Yes	Ville	Paris
formatted_address	string		Adresse complète <i>Si non fourni, ce champs sera construit avec les informations renseignées dans les autres champs</i>	61 Rue de Cléry 75 002 Paris France
google_id	string		Identifiant google de l'adresse	ChIJQTUM5BZu5kcR4Z7uOkT8CMk
location	Location		Coordonnées GPS de l'adresse	[48.8689952, 2.349442800000002]
address_more	string		Complément d'adresse	
floor	string		Étage	Rez-de-chaussée
door	string		Porte	101
building	string		Bâtiment	A
code	string		Digicode	75A

## Entité: CustomersSearch

`CustomersSearch` représente un client résultant de l'endpoint `/customers/search``, les données sont allégées par rapport à l'Entité: **Customer**.

### Exemple

```
{
  "id": 123456,
  "uuid": "...",
  "remote_id": "remote_123456",
  "nice_name": "Toto Test",
  "card": "abcd98765",
  "phone": "+33612345678",
  "mail": "toto.test@zelty.fr"
}
```

### ATTRIBUTS

Nom	Type	Description	Exemple
id	int	ID Zelty	123456

remote_id	string	Identifiant distant	<i>remote_123456</i>
nice_name	string	Prénom & Nom	<i>Toto Test</i>
card	string	Numéro de carte de fidélité.	<i>abcd98765</i>
phone	string	Numéro de téléphone. <i>Format international E164</i>	<i>+33612345678</i>
mail	string	Email du client	<i>toto.test@zelty.fr</i>

# Les commandes

## GET /orders

### Exemple de requête

```
curl https://api.zelty.fr/2.9/orders?from=YYYY-MM-DD&to=YYYY-MM-DD \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "orders": [
    {
      "id": 123123,
      "uuid": "000033-1312321-123123-12312",
      "comment": null,
      "device_id": 47553,
      "closed_by_device_id": 47553,
      "remote_id": null,
      "ref": "t44",
      "loyalty": 0,
      "seats": 5,
      "table": 10,
      "id_restaurant": 49,
      "created_at": "2019-08-01T16:18:57+02:00",
      "closed_at": "2019-08-01T16:48:57+02:00",
      "due_date": null,
      "mode": "eat_in",
      "source": "web",
      "status": "opened",
      "price": {
        "final_amount_inc_tax": 555,
        "final_amount_exc_tax": 555,
        "discounts": [],
        "taxes": []
      },
      "transactions": [
        { ... }
      ],
      "items": [
        { ... }
      ]
    }
  ]
}
```

```
    ],
    "customer": null,
    "channel": "Zelty Caisse",
    "virtual_brand_name": null,
    "first_name": "toto",
    "user": "Julien"
  }
}
```

Récupération de la liste des commandes dans un interval de temps donné

PARAMÈTRES GET

La requête doit se faire obligatoirement avec l'un des paramètres suivants :

- Soit les paramètres `from` et `to`.
- Soit le paramètre `noz`.
- Soit le paramètre `opened`.

Paramètre	Type	Description	Exemple
from	date	A partir de ... Format <code>YYYY-MM-DD</code> .	?from=2024-07-01
to	date	Jusqu'à ... Format <code>YYYY-MM-DD</code> . <i>Maximum : 31 jours.</i>	?to=2024-07-31
include_cancelled	bool	Permet d'ajouter les commandes annulées dans les résultats de <i>from</i> & <i>to</i>	?include_cancelled=1
noz	bool	Avec ce paramètre, il est possible de récupérer les commandes cloturées et qui n'ont pas encore subi le Z de caisse. Incompatible avec <i>from</i> & <i>to</i>	?noz=1
opened	bool	Avec ce paramètre, il est possible de récupérer les commandes ouvertes/en attentes. Incompatible avec <i>from</i> & <i>to</i>	?opened=1
expand	string[]	Voir <b>Champs complémentaires</b>	?expand[]=user
limit	int	Récupérer au maximum <code>X</code> commandes. Par défaut, 100. Maximum 200.	100
offset	int	Récupérer les commandes à partir de la <code>X</code> ème	0
till_id	int	Récupérer les commandes liées à la <b>closures.id</b>	1234
mode	string	Récupérer les commandes liées au mode choisi <i>eat_in</i> = sur place <i>takeaway</i> = à emporter <i>delivery</i> = en livraison	<i>eat_in</i>

AUTRES PARAMÈTRES GET

Ces paramètres sont à rentrer avec l'endpoint : `/order`

Paramètre	Type	Description	Exemple
table	int	Permet de récupérer la commande du n° de la table indiquée	7
ref	string	Permet de récupérer la commande du n° de ticket indiqué	t42

## RÉPONSE

Nom	Type	Description	Exemple
id	int	ID Zelty	44
id_restaurant	int	ID Zelty du restaurant	46
remote_id	string	Identifiant distant du restaurant	m47
ref	string	Numéro de ticket (read-only)	t44
created_at	datetime	Date de création de la commande <i>Date au format ISO 8601</i>	2019-08-01T16:18:57+02:00
closed_at	datetime	Date de clôture de la commande <i>Date au format ISO 8601</i>	2019-08-01T16:48:57+02:00
due_date	datetime	Date de livraison/récupération <i>Date au format ISO 8601</i>	2017-04-28T19:18:49+00:00
mode	string	Mode de commande <i>eat_in = sur place</i> <i>takeaway = à emporter</i> <i>delivery = en livraison</i>	eat_in
source	string	Provenance de la commande <i>pos</i> <i>remote</i> <i>web</i> <i>ubereats...</i>	web
price	objet	Tableau associatif des prix de la commande	
price.final_amount_inc_tax	int	Montant TTC de la commande	555
price.final_amount_exc_tax	int	Montant HT de la commande. <i>Disponible après la clôture de la commande</i>	555
price.discounts	array	Liste des remises globales (hors remises produit) <i>Non visible par défaut</i> <i>Voir <b>Champs complémentaires</b></i>	
price.taxes	array	Liste des taxes (dispo après clôture de la commande) <i>Non visible par défaut</i> <i>Voir <b>Champs complémentaires</b></i>	
seats	int	Nombre de couverts	5

table	int	Numéro de table <i>Uniquement pour les commandes sur place</i>	10
customer	Customer	Client lié à la commande <i>Non visible par défaut</i> Voir <b>Champs complémentaires</b>	Voir <b>Entité: Customer</b>
address	Address	Adresse de livraison <i>Uniquement si c'est une livraison.</i> <i>Factultatif si <code>delivery_address_id</code> est renseigné.</i>	Voir <b>Entité: Address</b>
transactions	Transaction[]	Liste des paiements déjà effectués sur la commande <i>Non visible par défaut</i> Voir <b>Champs complémentaires</b>	Voir <b>Entité: Transaction</b>
items		Liste des produits/menus dans la commande <i>Non visible par défaut</i> Voir <b>Champs complémentaires</b>	Voir <b>Entité: OrderEntry (GET)</b>
comment	string	Commentaire libre sur la commande	
device_id	id	ID de l'appareil qui a créé la commande	47553
first_name	string	Prénom attribué à la commande	Toto
closed_by_device_id	id	ID de l'appareil qui a fermé la commande	47553
user	string	Nom du serveur de la commande <i>Non visible par défaut</i> Voir <b>Champs complémentaires</b>	Zelty
delivery_started	datetime	Date de début de livraison <i>Date au format ISO 8601</i> <i>visible seulement si la commande est de type livraison</i>	2017-04-28T19:18:49+00:00
delivery_ended	datetime	Date de fin de livraison <i>Date au format ISO 8601</i> <i>visible seulement si la commande est de type livraison</i>	2017-04-28T19:48:49+00:00

## GET /orders/:id

Exemple de requête

```
curl https://api.zelty.fr/2.9/orders/<id> \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
```

```
"order": {
  "id": 123123,
  "uuid": "000033-1312321-123123-12312",
  "comment": null,
  "device_id": 12345,
  "remote_id": null,
  "ref": "t44",
  "loyalty": 0,
  "seats": 0,
  "table": 0,
  "id_restaurant": 49,
  "created_at": "2021-02-10T15:00:00+01:00",
  "due_date": null,
  "mode": "eat_in",
  "source": "web",
  "status": 255,
  "price": 555,
  "taxes": [
    { ... }
  ],
  "items": [
    {...}
  ]
}
```

Récupération des détails d'une commande

PARAMÈTRES GET

Paramètre	Type	Description	Exemple
expand	string[]	Voir <b>Champs complémentaires</b>	?expand[]=user

RÉPONSE

Voir Réponse de **GET /orders**

## POST /orders

Exemple de requête

```
curl https://api.zelty.fr/2.9/orders \
-H 'Authorization: Bearer <token_api>'
```

Body

```
// Commande sur place
{
  "id_restaurant": 42,
  "mode": "eat_in",
  "source": "web",
  "display_id": "89BE",
  "remote_id": "74a9246e-de3c-4f7d-b76d-6a484d575dac",
  "virtual_brand_name": "Virtual Chicken",
```

```

    "due_date": "2019-04-25T05:00:00+02:00",
    "comment": "Commentaire au niveau de la commande",
    "customer":
      {
        "name": "toto",
        "mail": "test.toto@zelty.fr"
      },
    "items": [
      {
        "type": "dish",
        "id": 602508,
        "comment": "Commentaire au niveau du produit",
        "place": 2
      },
      {
        "type": "menu",
        "id": 59504,
        "dishes": [
          {
            "id_part": 94629,
            "id": 1307412,
            "comment": "Commentaire au niveau du produit dans le menu",
            "modifiers": [
              {
                "option_id": 210302,
                "option_value_id": 1049377,
                "quantity": 1,
                "price": 50
              }
            ]
          }
        ]
      }
    ]
  }
}

// Commande en livraison
{
  "id_restaurant": 42,
  "mode": "delivery",
  "source": "web",
  "display_id": "AZ12",
  "remote_id": "74a9246e-de3c-4f7d-b76d-6a484d575dac",
  "virtual_brand_name": "Virtual Chicken",
  "due_date": "2019-04-25T05:00:00+02:00",
  "comment": "Commentaire au niveau de la commande",
  "customer":
    {
      "name": "toto",
      "mail": "test.toto@zelty.fr"
    },
  "items": [{
    "type": "dish",
    "id": 206182,
    "place": 2,
    "comment": "Commentaire au niveau du produit"
  }
]
}

```

## Réponse

```
{
  "errno": 0
}
```

Création d'une nouvelle commande

BODY

Nom	Type	Requis	Description	Exemple
remote_id	string		Identifiant distant (unique par établissement, max 128 caractères)	<i>m43</i>
display_id	string		Numéro de commande pour le client (si non fourni, le numéro de commande Zelty sera utilisé, max 10 caractères)	<i>89EB</i>
id_restaurant	int	Yes	ID Zelty du restaurant	<i>46</i>
restaurant_remote_id	string		Identifiant distant du restaurant	<i>m47</i>
due_date	datetime	Yes	Date de livraison/récupération <i>Date au format ISO 8601</i>	<i>2017-04-28T19:18:49+00:00</i>
source	string		Source de la commande : web mobile kiosk	<i>web</i>
virtual_brand_name	string		Nom de la marque virtuelle (uniquement dans le cas où le restaurant gère plusieurs marques)	<i>Virtual Chicken</i>
mode	string	Yes	Mode de commande <i>eat_in = sur place</i> <i>takeaway = à emporter</i> <i>delivery = en livraison</i>	<i>eat_in</i>
table	int		Numéro de table <i>Uniquement pour les commandes sur place</i>	<i>10</i>
seats	int		Nombre de couverts <i>Uniquement pour les commandes sur place</i>	<i>2</i>
buzzer_ref	string		Numéro du buzzer <i>(10 caractères maximum)</i>	<i>Buzzer 42</i>
customer	Customer		Client lié à la commande. <i>Seul <code>id</code> / <code>remote_id</code> / <code>mail</code> / <code>phone</code> du client est **obligatoire, s'il existe déjà.</i> <i>Si d'autres valeurs sont présentes la fiche client sera mise à jour.</i> <i>Pour une commande en livraison, ce champ est <b>obligatoire</b>.</i>	<b>Voir Entité: Customer</b>
			Adresse de livraison <i>Uniquement si c'est une livraison.</i>	<b>Voir Entité:</b>



address	Address	Yes	Factultatif si <code>delivery_address_id</code> est renseigné	Address
id_delivery_address	int	Yes	ID Zelty de l'adresse de livraison (liée au client) Uniquement si c'est une livraison Factultatif si <code>address</code> est renseigné	45
id_delivery_zone	int	Yes	ID Zelty de la zone de livraison associée Uniquement si c'est une livraison Facultatif si la zone peut être déduite de l'adresse	45
items	OrderEntry[]	Yes	Liste des produits et menus	
id_promotion	int		ID de la promotion liée	4200
promotion_discount	int		Montant de la remise promotionnelle (si elle s'applique sur le total de la commande)	1000
total	int		Total de la commande Si le total inférieur au total calculé par Zelty une remise de la différence sera appliquée	
transactions	Transaction[]		Liste des paiements déjà effectués sur la commande	Voir <b>Entité: Transaction</b>
needed_transactions	int[]		Liste des méthodes de paiements nécessaires à la livraison	
comment	string		Commentaire libre sur la commande (256 caractères maximum)	
first_name	string		Permet d'attribuer un prénom à la commande, sans créer de fiche client	Zelty
phone	string		Numéro de téléphone. Format international E164	+33123456789

RÉPONSE

N/A

PATCH /orders/:id

Exemple de requête

```
curl https://api.zelty.fr/2.9/orders/<id> \
-H 'Authorization: Bearer <token_api>'
```

Body

```
{
  "comment": "Modification du commentaire de la commande",
  "customer": {
    "id": 123456
  },
  "items": [
    {
      "id": 602508
    }
  ]
}
```

Réponse

```
{
  "errno": 0
}
```

Ajout des produits dans une commande

PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int	Identifiant de la commande	42

BODY

Paramètre	Type	Description	Exemple
comment	string	Modifier le commentaire de la commande ouverte	Ajout de couvert
customer	Customer	Permet d'associer un client à la commande, si aucun n'est associé encore	Voir <b>Entité: Customer</b>
items	OrderEntry[]	Ajout de produit(s)	Voir <b>Entité: OrderEntry (POST)</b>

NB : Seul une commande ouverte est modifiable.

RÉPONSE

N/A

## POST /orders/:id/transactions

Exemple de requête

```
curl https://api.zelty.fr/2.9/orders/<id>/transactions \
-H 'Authorization: Bearer <token_api>'
```

### Body

```
{
  "transactions": [
    {
      "name": "CB",
      "price": 1000
    }
  ],
  "close_if_paid": true
}
```

### Réponse

```
{
  "errno": 0
}
```

Ajout d'une/de plusieurs méthode(s) de paiement existante(s) dans une commande ouverte

#### PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int	Identifiant de la commande	42

#### BODY

Paramètre	Type	Requis	Description	Exemple
transactions	Transaction[]	YES	Ajout des méthodes de paiement	Voir <b>Entité: Transaction</b>
close_if_paid	Bool		Clôture la commande si le total est exact	true

#### RÉPONSE

N/A

## POST /orders/:id/closure

Exemple de requête

```
curl https://api.zelty.fr/2.9/orders/<id>/closure \
-H 'Authorization: Bearer <token_api>'
```

### Body

### Réponse

```
{
  "errno": 0
}
```

*La requête permet de clôturer une commande éligible à la clôture automatique.  
Pour qu'une commande soit éligible, elle doit posséder des produits et être soldée.*

#### PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int	Identifiant de la commande	42

#### BODY

N/A

#### RÉPONSE

N/A

## POST /orders/:id/production

Exemple de requête

```
curl https://api.zelty.fr/2.9/orders/<id>/production \
-H 'Authorization: Bearer <token_api>'
```

#### Body

```
{
  "entry_ids": [123,124],
  "subentry_ids": [223,224]
}
```

#### Réponse

```
{
  "errno": 0
}
```

*Envoi en production d'un/de plusieurs produit(s) existante(s) dans une commande ouverte.*

**Attention :** l'option "Autoriser l'envoi en production depuis l'extérieur depuis cet appareil" doit être activée sur Zelty Caisse.

#### PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int	Identifiant de la commande	42

## BODY

Paramètre	Type	Description	Exemple
entry_ids	int[]	Liste des ID de la ligne à envoyer en production. Ex: Un menu, un produit. <code>items.id</code>	[123,124]
subentry_ids	int[]	Liste des ID de la sous-ligne à envoyer en production. Ex : Un produit dans un menu. <code>items.modifiers.id</code>	[223,224]

## RÉPONSE

N/A

## Champs complémentaires

`expand` est un paramètre permettant de récupérer des informations complémentaires lors de certaines requêtes GET, sur certains endpoints.

## LISTE DES ENDPOINTS CONCERNÉS :

- GET /orders
- GET /order/:id (page à venir)

## LISTE DES PROPRIÉTÉS

Exemple de requête

```
curl https://api.zelty.fr/2.9/orders/<id>?expand[]=transactions&expand[]=transactions.method \
-H 'Authorization: Bearer <token_api>'
```

## Réponse

```
{
  "order": {
    ...,
    "transactions": [
      {
        "device_id": 63,
        "price": 1610,
        "date": "2020-11-05T14:02:02+01:00",
        "method": "Carte Bleue"
      }
    ]
  },
  "errno": 0
}
```

Toutes les propriétés peuvent être utilisées indépendamment au besoin, ou bien cumulées ensemble si nécessaire.

NB : `transactions.method` s'utilise obligatoirement avec `transactions`.

Propriété	Type	Description	Exemple
-----------	------	-------------	---------

user	string	Le nom du serveur	Manager
customer	Customer	Détail du client	Voir <b>Entité: Customer</b>
deliveryman	Deliveryman	Informations sur le livreur	Voir <b>Entité: Deliveryman</b>
delivery_address	Address	L'adresse de livraison	Voir <b>Entité: Address</b>
transactions	OrderTransaction[]	Informations sur le paiement	Voir <b>Entité: OrderTransaction</b>
transactions.method	string	Nom de la méthode de paiement	Carte Bleue
items	array	Liste des produits	Voir <b>Entité: OrderEntry (GET)</b>
price.taxes	array	Liste des taxes (dispo après clôture de la commande)	
price.discounts	array	Liste des remises globales (hors remises produit)	

## Entité: OrderEntry (GET)

### Exemple

```
{
  "name": "Frites",
  "type": "dish",
  "item_id": "42",
  "price": {
    "base_original_amount_inc_tax": 1600,
    "original_amount_inc_tax": 1600,
    "discounted_amount_inc_tax": 600,
    "final_amount_inc_tax": 0,
    "tax": {
      "tax_amount": 0,
      "tax_rate": null
    }
  },
  "modifiers": [
    {
      "id": 43,
      "type": "option_value",
      "name": "Ketchup",
      "quantity": 1,
      "price": {
        "original_amount_inc_tax": 0
      }
    }
  ]
}
```

**OrderEntry** (GET) représente une ligne lors de la récupération d'une commande.

ATTRIBUTS

Nom	Type	Description	Exemple
name	string	Nom du produit/menu	<i>Frites</i>
type	string	Plat ou menu	<i>dish/menu</i>
item_id	string	ID du produit/menu	<i>42</i>
price	object		<i>**</i>
price.base_original_amount_inc_tax	int	prix de la ligne (hors remise et hors suppléments/options)	<i>1600</i>
price.original_amount_inc_tax	int	prix de la ligne avec suppléments/options	<i>1600</i>
price.discounted_amount_inc_tax	int	prix de la ligne avec suppléments/options et potentielle remise	<i>600</i>
price.final_amount_inc_tax	int	prix final de la ligne incluant aussi les remises globales de la commande (dispo uniquement après clôture)	<i>0</i>
tax	object		<i>**</i>
tax.tax_amount	int	montant de la TVA (dispo uniquement après clôture)	<i>0</i>
tax.tax_rate	int/null	taux de TVA dans le cas d'un taux unique. Dans le cas d'un menu avec plusieurs taux cette valeur sera nulle	<i>null</i>
modifiers	array		<i>**</i>
modifiers[].id	int		<i>43</i>
modifiers[].type	string		<i>option_value</i>
modifiers[].name	string		<i>Ketchup</i>
modifiers[].quantity	int		<i>1</i>
modifiers[].price	object		<i>**</i>
modifiers[].price.original_amount_inc_tax	int		<i>0</i>

Entité: OrderEntry (POST)

Exemple

```
// exemple si type = dish
```

```
{
  "type": "dish",
  "id": 42,
  "comment": "Commentaire sur le Produit",
  "modifiers": [
    {
      "option_id": 43,
      "option_value_id": 44,
      "quantity": 1,
      "price": 100
    }
  ],
  "price": 1000
}

// exemple si type = menu
{
  "type": "menu",
  "id": 142,
  "dishes": [
    {
      "id_part": 442,
      "id": 42,
      "comment": "Commentaire sur le Produit dans le Menu",
      "modifiers": [
        {
          "option_id": 43,
          "option_value_id": 44,
          "quantity": 1,
          "price": 100
        }
      ]
    }
  ],
  "price": 1500
}
```

`OrderEntry` (POST) représente une ligne lors de l'envoi d'une commande.

ATTRIBUTS

Nom	Type	Requis	Description	Exemple
type	string	Yes	Plat ou menu	dish/menu
id	int	Yes	ID Zelty du plat ou menu	42
remote_id	string		Identifiant distant du plat ou menu	m42
modifiers	OrderModifier[]		Uniquement quand <b>type = dish</b> Valeurs des options du plat La quantité maximum de tous les <code>modifiers</code> d'un produit ne peut pas excéder 20.	Voir <b>Entité: OrderModifier</b>
dishes	OrderEntryMenuPart[]		Uniquement quand <b>type = menu</b> Liste des produits qui composent le menu La quantité maximum de tous les produits d'un menu ne peut pas	Voir <b>Entité: OrderEntryMenuPart</b>



		excéder 20.	
price	int	Prix de la ligne (suppléments compris)	1000
promotion_discount	int	Remise liée à la promotion de la commande. La remise est prise en compte <b>seulement</b> si le champs <code>id_promotion</code> est présent et valide dans la commande.	1000
place	int	Indique la place du produit dans la commande. La valeur va de 1 à 10	1
comment	string	<i>Uniquement quand <b>type = dish</b></i> Commentaire libre sur le produit. (140 caractères maximum)	

## Entité: OrderModifier

### Exemple

```
// Exemple avec les ID Zelty
{
  "option_id": 42,
  "option_value_id": 43,
  "quantity": 1,
  "price": 0
}

// Exemple avec les ID distants
{
  "option_remote_id": "remote_42",
  "option_value_remote_id": "remote_43",
  "quantity": 1,
  "price": 0
}
```

`OrderModifier` représente les options choisies pour un produit.

**Attention :** Un produit ne peut pas avoir plus de 20 `modifiers` lors de l'envoi d'une commande.

### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
option_id	int	Yes	ID du groupe d'options	42
option_remote_id	string		ID distant du groupe d'options	remote_42
option_value_id	int	Yes	ID de la valeur	43

option_value_remote_id	string		ID distant de la valeur	<i>remote_43</i>
quantity	int	<b>Yes</b>	quantité de l'option	<i>1</i>
price	int	<b>Yes</b>	prix de l'option	<i>0</i>

## Entité: OrderEntryMenuPart

### Exemple

```
{
  "id_part": 42,
  "id": 43,
  "modifiers": [{...}],
  "comment": "Sans cornichon"
}
```

`OrderEntryMenuContent` représente le contenu d'un menu dans une commande.

### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id_part	int	<b>Yes</b>	ID Zelty du MenuPart	<i>42</i>
part_remote_id	string		Identifiant distant du MenuPart	<i>m42</i>
id	int	<b>Yes</b>	ID Zelty du plat ou menu	<i>43</i>
remote_id	string		Identifiant distant du plat ou menu	<i>m43</i>
modifiers	OrderModifier[]		Valeurs des options du plat <i>Uniquement quand type = dish</i>	Voir <b>Entité: OrderModifier</b>
comment	string		Commentaire sur le produit dans le menu	<i>Sans cornichon</i>

## Entité: OrderStatus

### Exemple

```
{
  "id": 42,
  "uuid": "00000-00000-abcde",
  "remote_id": "m43",
  "status": "ended",
  "delivery_status": "delivered"
}
```

`OrderStatus` représente le statut d’une commande en cours. Tous les champs sont “read-only”.

ATTRIBUTS

Nom	Type	Description	Exemple
id	int	ID Zelty	42
remote_id	string	Identifiant distant	m43
status	string	Statut de la commande : - 'opened' : commande ouverte - 'cancelled' : commande annulée - 'ended' : commande payée et terminée	ended
delivery_status	string	Statut de la livraison : - 'pending' : en attente de livraison - 'delivering' : en cours de livraison - 'delivered' : livré	delivering

## Entité: Transaction

Exemple

```
{
  "transactions": [
    {
      "id_transaction_method": 1234,
      "price": 1000
    }
  ]
}
```

`Transaction` représente un paiement.

ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id_transaction_method	int	Yes*	ID Zelty de la méthode de paiement	1234
remote_id	string		Id distant unique de la méthode de paiement. Seul le Format UUID v4 est accepté.	abcdef-1234-1234-1234-azertyuiop
name	string	Yes*	Nom de la méthode de paiement	CB
price	int	Yes**	Montant de la transaction	1000

\*Seul l'un des 2 champs (`id_transaction_method` ou `name`) est requis.

\*\*La valeur de `price` ne peut pas être à 0.

\*\*\*`remote_id` est une propriété optionnelle, si l'on renseigne une valeur déjà existante, la transaction sera ignorée, sans

retour de message d'erreur.

## Entité: Deliveryman

### Exemple

```
{
  "id" : 42,
  "name" "Toto"
}
```

Retourne en complément un objet `deliveryman` lié à la commande

Paramètre	Type	Description	Exemple
id	int	ID du livreur	42
name	String	Nom du livreur	Toto

## Entité: OrderTransaction

### Exemple

```
{
  "device_id": 42,
  "price": 1610,
  "date": "2020-11-05T14:02:02+01:00",
  "id_transaction_method": 142,
  "method": "Carte Bleue"
}
```

Retourne en complément un objet `OrderTransaction` lié à la commande

Paramètre	Type	Description	Exemple
device_id	int	ID de l'appareil	42
price	int	Montant de la transaction (1610 = 16€10)	1610
date	datetime	Date de la transaction	2020-11-05T14:02:02+01:00
id_transaction_method	int	ID de la méthode de paiement	142
method	string	Nom de la méthode de paiement	Carte Bleue

## Entité: Tax

### Example

```
{
  "rate": 1000,
  "name": "10.00%",
  "amount": 146,
  "inc_tax": 1610,
  "exc_tax": 1464
}
```

Retourne en complément une liste d'objet `tax` liée à la commande

Paramètre	Type	Description	Exemple
rate	int	Pourcentage de la taxe. 1000 pour 10%	1000
name	string	Nom de la taxe	"10.00%"
amount	int	Montant de TVA	146
inc_tax	int	Montant TTC	1610
exc_tax	int	Montant HT	1464

## Les clôtures de caisse

### GET /closures

#### Exemple de requête

```
curl https://api.zelty.fr/2.9/closures \
-H 'Authorization: Bearer <token_api>'
```

#### Réponse

```
{
  "closures": [
    {
      "id": 123456,
      "id_restaurant": 42,
      "created_at": "2020-12-31T18:19:55+0200",
      "date": "2020-12-31",
      "turnover": 452312,
      "comment": "Commentaire sur la clôture",
      "taxes": 48462
    }
  ]
}
```

```
    }
  }
}
```

Liste des clôtures de caisse

PARAMÈTRES GET

Nom	Type	Description	Exemple
before	date	Récupérer les clôtures avant telle date Format <code>YYYY-MM-DD</code>	<code>2024-07-31</code>
after	date	Récupérer les clôtures après telle date Format <code>YYYY-MM-DD</code>	<code>2024-07-01</code>
limit	int	Récupérer au maximum <code>x</code> clôtures <i>Valeur max : 200</i>	100
offset	int	Récupérer les clôtures à partir de la <code>x</code> ème	0
restaurant_ids	string[]	Filtrer sur plusieurs restaurants	<code>?restaurant_ids[]=42</code>

RÉPONSE

Nom	Type	Description	Exemple
closures	Closure[]	Liste des clôtures	Voir <b>Entité: Closure</b>

## Entité: Closure

Exemple

```
{
  "id": 123456,
  "id_restaurant": 42,
  "created_at": "2020-12-31T18:19:55+0200",
  "date": "2020-12-31",
  "turnover": 452312,
  "comment": "Commentaire sur la cloture",
  "taxes": 48462
}
```

`Closure` représente une clôture de caisse. Les clôtures ne sont pas modifiables.

ATTRIBUTS

Nom	Type	Description	Exemple
id	int	ID Zelty	123456
id_restaurant	int	Identifiant restaurant	43

created_at	datetime	Date et heure de la clôture <i>Date au format ISO 8601</i>	2020-12-31T18:19:55+0200
date	date	Journée de la clôture	2020-12-31
turnover	int	Chiffre d'affaire TTC de la journée <i>Montant exprimé en centimes sans virgule</i>	452312
comment	string	Commentaire libre sur la clôture de caisse	Commentaire sur la cloture
taxes	int	Montant des taxes <i>Montant exprimé en centimes sans virgule</i>	48462

# La réservation

## GET /bookings

### Exemple de requête

```
curl https://api.zelty.fr/2.9/bookings \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "bookings": [
    {
      "id": 42,
      "uid": "00000-610cfccd-azerty",
      "remote_id": "m42",
      "id_customer": 44,
      "id_command": 45,
      "created_at": "2020-12-31T19:18:49+00:00",
      "booking_for": "2020-12-31T19:18:49+00:00",
      "arrived_at": null,
      "closed_at": null,
      "table": 15,
      "places": 3,
      "status": 0,
      "cancel_reason": 0,
      "src": "web",
      "comment": "Commentaire de la réservation",
      "final_price": null,
      "customer": {...}
    }
  ]
}
```

Récupération de la liste des réservations.

### PARAMÈTRES GET

Nom	Type	Description	Défaut
date	date	Récupérer les réservations de la date indiquée. Format <code>YYYY-MM-DD</code> <i>Par défaut : date du jour</i>	<code>2024-07-01</code>

RÉPONSE

Nom	Type	Description	Exemple
bookings	Booking[]	Liste des réservations	Voir <b>Entité: Booking</b>

## GET /bookings/:id

Exemple de requête

```
curl https://api.zelty.fr/2.9/bookings/<id_booking> \
-H 'Authorization: Bearer <token_api>'
```

Réponse

```
{
  "booking": [
    {
      "id": 42,
      "uid": "00000-610cfccd-azerty",
      "remote_id": "m42",
      "id_customer": 44,
      "id_command": 45,
      "created_at": "2020-12-31T19:18:49+00:00",
      "booking_for": "2020-12-31T19:18:49+00:00",
      "arrived_at": null,
      "closed_at": null,
      "table": 15,
      "places": 3,
      "status": 0,
      "cancel_reason": 0,
      "src": "web",
      "comment": "Commentaire de la réservation",
      "final_price": null,
      "customer": {...}
    }
  ]
}
```

Récupération d'une réservation.

PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int / string	<code>id</code> (int) ou <code>remote_id</code> (string) de la réservation	<code>42 / remote_42</code>



PARAMÈTRES GET

N/A

RÉPONSE

Nom	Type	Description	Exemple
booking	Booking	Détail de la réservation	Voir <b>Entité: Booking</b>

## POST /bookings

Exemple de requête

```
curl https://api.zelty.fr/2.9/bookings/ \
-H 'Authorization: Bearer <token_api>'
```

Body

```
{
  "id": 142,
  "remote_id": "r142",
  "id_restaurant": 42,
  "booking_for": "2024-09-03T12:00:00",
  "table": 12,
  "places": 3,
  "status": 0,
  "comment": "Commentaire de la réservation",
  "customer": {
    "id": 12345,
    "fname": "Ftoto",
    "name": "toto",
    "personal_info": ""
  }
}
```

Réponse

```
{
  "errno": 0
}
```

Création/modification d’une réservation

BODY

Le corps de la requête doit contenir un objet de type `BookingNew`.

PARAMÈTRES URL

Nom	Type	Description	Exemple
		Permet de vérifier l'existence de la fiche client :	

<code>check_existing</code>	<code>bool</code>	- via son email ( <code>mail</code> ) dans un premier temps, puis - via son n° de téléphone ( <code>phone</code> ) dans un second temps. Si le client existe : la requête va mettre à jour la fiche client, sinon la requête va créer une nouvelle fiche client.	<code>check_existing=1</code>
-----------------------------	-------------------	--	-------------------------------

## RÉPONSE

N/A

# POST /bookings/:id

## Exemple de requête

```
curl https://api.zelty.fr/2.9/bookings/<id_booking> \
-H 'Authorization: Bearer <token_api>'
```

## Body

```
{
  "id": 142,
  "remote_id": "r142",
  "id_restaurant": 42,
  "booking_for": "2024-09-03T12:00:00",
  "table": 12,
  "places": 3,
  "status": 0,
  "comment": "Commentaire de la réservation",
  "customer": {
    "id": 12345,
    "fname": "Ftoto",
    "name": "toto",
    "personal_info": ""
  }
}
```

## Réponse

```
{
  "errno": 0
}
```

*Modification d'une réservation existante.*

## PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int	Identifiant de la réservation	42

## BODY

Le corps de la requête doit contenir un objet de type `BookingNew`.

RÉPONSE

N/A

Entité: Booking

Exemple

```
{
  "id": 42,
  "remote_id": "m42",
  "id_customer": 44,
  "id_command": 45,
  "created_at": "2020-12-31T19:18:49+00:00",
  "booking_for": "2020-12-31T19:18:49+00:00",
  "arrived_at": null,
  "closed_at": null,
  "table": 15,
  "places": 3,
  "status": 0,
  "cancel_reason": 0,
  "src": "web",
  "comment": "Commentaire de la réservation",
  "final_price": null,
  "customer": {...}
}
```

**Booking** représente une réservation.

ATTRIBUTS

Nom	Type	Description	Exemple
id	int	ID Zelty	42
remote_id	string	ID distant	m42
id_customer	int	ID du client lié à la réservation	44
id_command	int	ID de la commande associée à la réservation une fois les clients sur place	45
created_at	datetime	Date et heure de création Date au format ISO 8601	2020-12-31T19:18:49+00:00
booking_for	datetime	Date et heure de réservation Date au format ISO 8601	2020-12-31T19:18:49+00:00
arrived_at	datetime	Date et heure d'arrivée du client Date au format ISO 8601	2020-12-31T19:18:49+00:00
closed_at	datetime	Date et heure de la clôture de la réservation Date au format ISO 8601	2020-12-31T20:18:49+00:00
table	int	Numéro de table lié à la réservation	15

places	int	Nombre de couverts	3
status	int	Status de la réservation 0 = en attente de confirmation 80 = confirmée 96 = client installé 192 = annulée 255 = terminée	0
cancel_reason	int	Status de l'annulation (uniquement si status = 192) 0 = noshow 16 = annulée par le client 32 = annulée par le restaurant	0
src	string	Source de la réservation	web
comment	string	Commentaire libre sur la réservation	Commentaire de la réservation
final_price	int	Montant du ticket après réservation	1050
customer	Customer	Fiche client	Voir <b>Entité: Customer</b>

## Entité: BookingNew

### Exemple

```
{
  "id": 142,
  "remote_id": "r142",
  "id_restaurant": 42,
  "booking_for": "2024-09-03T12:00:00",
  "table": 12,
  "places": 3,
  "status": 0,
  "comment": "Commentaire de la réservation",
  "customer": {
    "id": 12345,
    "fname": "Ftoto",
    "name": "toto",
    "personal_info": ""
  }
}
```

**BookingNew** représente les données à poster lors d’une création/modification de réservation.

### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int		ID Zelty Uniquement pour une modification	42

remote_id	string		ID distant	<i>m42</i>
id_restaurant	int		ID du restaurant associé à la réservation	<i>43</i>
created_at	datetime		Date et heure de création <i>Date au format ISO 8601</i>	<i>2020-12-31T19:18:49+00:00</i>
booking_for	datetime	<b>Yes</b>	Date et heure de réservation <i>Date au format ISO 8601</i>	<i>2020-12-31T19:18:49+00:00</i>
customer	Customer	<b>Yes</b>	Client <i>Seul l'id ou le remote_id du client est obligatoire s'il existe déjà. Si d'autres valeurs sont présentes le client sera mis à jour.</i>	Voir <b>Entité: Customer</b>
table	int		Numéro de table lié à la réservation	<i>15</i>
places	int	<b>Yes</b>	Nombre de couverts	<i>3</i>
status	int	<b>Yes</b>	Status de la réservation <i>0 = en attente de confirmation 80 = confirmée 96 = client installé 192 = annulée 255 = terminée</i>	<i>0</i>
cancel_reason	int		Status de l'annulation (uniquement si status = 192) <i>0 = noshow 16 = annulée par le client 32 = annulée par le restaurant</i>	<i>0</i>
comment	string		Commentaire	<i>Commentaire de la réservation</i>

## Le plan de salle

### GET /tableMap/rooms

#### Exemple de requête

```
curl https://api.zelty.fr/2.9/tableMap/rooms \
-H 'Authorization: Bearer <token_api>'
```

#### Réponse

```
{
  "rooms": [
    {
      "id": 3,
```

```
    "name": "Bar",
    "tables": [
      {
        "id": 1,
        "num": 1,
        "figure": 0,
        "pos": [
          130,
          392
        ],
        "size": [
          1,
          1
        ]
      },
      {
        "id": 20,
        "num": 20,
        "figure": 1,
        "pos": [
          273,
          512
        ],
        "size": [
          1,
          1
        ]
      }
    ],
    "decorations": [
      {
        "id": 1,
        "color": null,
        "pos": [
          705,
          289
        ],
        "size": [
          83,
          5
        ]
      }
    ]
  },
  ]
}
```

Récupération des salles

PARAMÈTRES GET

N/A

RÉPONSE

Nom	Type	Description	Exemple
rooms	Room[]	Détail des salles	Voir <i>Entité: Room</i>

# GET /tableMap/rooms/:id

## Exemple de requête

```
curl https://api.zelty.fr/2.9/tableMap/rooms/<id> \
-H 'Authorization: Bearer <token_api>'
```

## Réponse

```
{
  "room": {
    "id": 3,
    "name": "Bar",
    "tables": [
      {
        "id": 1,
        "num": 1,
        "figure": 0,
        "pos": [
          130,
          392
        ],
        "size": [
          1,
          1
        ]
      },
      {
        "id": 20,
        "num": 20,
        "figure": 1,
        "pos": [
          273,
          512
        ],
        "size": [
          1,
          1
        ]
      }
    ],
    "decorations": [
      {
        "id": 1,
        "color": null,
        "pos": [
          705,
          289
        ],
        "size": [
          83,
          5
        ]
      }
    ]
  }
}
```

Récupération d'une salle avec son id

PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int	Identifiant de la salle	42

PARAMÈTRES GET

N/A

RÉPONSE

Nom	Type	Description	Exemple
rooms	Room	Détail de la salle	Voir <b>Entité: Room</b>

## Entité: Room

Exemple

```
{
  "id": 42,
  "name": "Bar",
  "tables": [
    {
      "id": 1,
      "num": 1,
      "figure": 0,
      "pos": [
        130,
        392
      ],
      "size": [
        1,
        1
      ]
    }
  ],
  "decorations": [
    {
      "id": 1,
      "color": null,
      "pos": [
        705,
        289
      ],
      "size": [
        83,
        5
      ]
    }
  ]
}
```



**Room** représente une salle dans le restaurant.

Paramètre	Type	Description	Exemple
id	int	ID Zelty	42
name	string	Nom de la salle	Bar
tables	Table[]	Informations sur les tables	Entite Table ?
decorations	Decoration[]	Informations sur les décorations	Entite Décoration ?

## Entité: Table

### Exemple

```
{
  "id": 1,
  "num": 1,
  "figure": 0,
  "pos": [
    130,
    392
  ],
  "size": [
    1,
    1
  ]
}
```

**Table** représente une table dans le restaurant.

### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int	Yes	ID Zelty	1
num	int	Yes	Numéro de la table	1
figure	int	Yes	Forme de la table. 0 = forme rectangulaire 1 = forme ronde	0
pos	int[]	Yes	Position de la table	[130,392]
size	int[]	Yes	Taille de la table	[1,1]

# Entité: Decoration

## Exemple

```
{
  "id": 1,
  "color": "#946b6b",
  "pos": [
    705,
    289
  ],
  "size": [
    83,
    5
  ]
}
```

`Decoration` représente une décoration dans le restaurant.

### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int	Yes	ID Zelty	1
color	string	Yes	Couleur de la décoration <i>Format Hexadecimal</i>	#946b6b
pos	int[]	Yes	Position de la table	[705,289]
size	int[]	Yes	Taille de la table	[83,5]

# Les promotions

## GET /coupons

### Exemple de requête

```
curl https://api.zelty.fr/2.9/coupons \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "coupons": [
    {
      "id": 123123,
      "code": "TotoTata",
    }
  ]
}
```

```
{
  "name": "Promo Toto",
  "valid_from": "2019-08-01T07:00:00+02:00",
  "valid_until": "2019-08-08T23:30:00+02:00",
  "min_price": 0,
  "promo_type": "items",
  "required_items": null,
  "only_once": false,
  "first_command": false,
  "max_uses": 0,
  "modeflag": 7,
  "noweb": false,
  "nopus": false,
  "enable_hours": null,
  "repeat": false,
  "priority": 1,
  "disable": false,
  "ignore_order_min_price": false,
  "force_web_payment": false,
  "customer_registered_days": null,
  "uses": 5,
  "fixed_discount": null,
  "percent_discount": 1000,
  "promo_data": {
    "items_groups": [
      [
        { ... }
      ]
    ]
  }
}
```

### Récupération de la liste des promotions

Cela ne renvoie que les coupons encore valides et automatiques

#### PARAMÈTRES GET

Nom	Type	Description	Exemple
limit	int	Max = 500 Défaut = 100	<i>limit=50</i>
offset	int	Ne prend pas en compte les premiers résultats	<i>offset=51</i>
min_price	int	Retourne toutes les promos qui ont un montant minimum ou égale à cette valeur	<i>500</i>
customer_id	int	Retourne toutes les promos qui sont valides pour ce client	<i>**</i>

#### RÉPONSE

Nom	Type	Description	Exemple
coupons	Coupon[]	Liste des promotions	Voir <b>Entité: Promotion</b>

## GET /coupons/:id

Exemple de requête

```
curl https://api.zelty.fr/2.9/coupons/<id> \
-H 'Authorization: Bearer <token_api>'
```

Réponse

```
{
  "id": 123123,
  "code": "TotoTata",
  "name": "Promo Toto",
  "valid_from": "2019-08-01T07:00:00+02:00",
  "valid_until": "2019-08-08T23:30:00+02:00",
  "min_price": 0,
  "promo_type": "items",
  "required_items": null,
  "only_once": false,
  "first_command": false,
  "max_uses": 0,
  "modeflag": 7,
  "noweb": false,
  "nopus": false,
  "enable_hours": null,
  "repeat": false,
  "priority": 1,
  "disable": false,
  "ignore_order_min_price": false,
  "force_web_payment": false,
  "customer_registered_days": null,
  "uses": 5,
  "fixed_discount": null,
  "percent_discount": 1000,
  "promo_data": {
    "items_groups": [
      [
        { ... }
      ]
    ]
  }
}
```

Récupération d'une promotion avec son id

PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int	ID du coupon (promotion)	123123

PARAMÈTRES GET

Nom	Type	Description	Exemple
is_valid	bool	retourne la promotion que si elle est encore valide	<i>is_valid=1</i>
customer_id	int	permet de checker la validité sur le client en question	<i>customer_id=42</i>

RÉPONSE

Nom	Type	Description	Exemple
coupon	Coupon	Détail d'une promotion	Voir <i>Entité: Promotion</i>

## GET /coupon

### Exemple de requête

```
curl https://api.zelty.fr/2.9/coupon?code=<code> \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "coupon": {
    "id": 123123,
    "code": "TotoTata",
    "name": "Promo Toto",
    "valid_from": "2019-08-01T07:00:00+02:00",
    "valid_until": "2019-08-08T23:30:00+02:00",
    "min_price": 300,
    "promo_type": "total",
    "required_items": null,
    "only_once": true,
    "first_command": false,
    "max_uses": 0,
    "modelflag": 7,
    "noweb": false,
    "nopus": false,
    "enable_hours": null,
    "repeat": false,
    "disable": false,
    "ignore_order_min_price": false,
    "force_web_payment": false,
    "customer_registered_days": null,
    "uses": 0,
    "fixed_discount": null,
    "percent_discount": null,
    "promo_data": {
      "items_exclude": []
    }
  }
}
```

Récupération d'une promotion avec le 'code' du code promo

#### PARAMÈTRES GET

Nom	Type	Requis	Description	Exemple
code	string	Yes	Retourne le détail de la promo avec son code	
limit	int		Max = 500 Défaut = 100	<i>limit=50</i>

offset	int	Ne prend pas en compte les premiers résultats	<i>offset=51</i>
min_price	int	Retourne toutes les promos qui ont un montant minimum inférieur à cette valeur	<i>500</i>
customer_id	int	Retourne toutes les promos qui sont valides pour ce client	<i>**</i>

## RÉPONSE

Nom	Type	Description	Exemple
coupon	Coupon	Détail d'une promotion	Voir <b>Entité: Promotion</b>

# POST /coupons

## Exemple de requête

```
curl https://api.zelty.fr/2.9/coupons \
-H 'Authorization: Bearer <token_api>'
```

## Body

```
{
  "name": "Promotion Test",
  "promo_type": "total",
  "fixed_discount": 500,
  "percent_discount": 1000,
  "valid_from": "2022-01-01 10:30:00",
  "valid_until": "2022-12-31 10:30:00",
  "min_price": 1500,
  "disable": false,
  "noweb": true,
  "nopus": false,
  "ignore_order_min_price": false,
  "force_web_payment": false,
  "only_once": true,
  "customer_registered_days": 10,
  "max_uses": 1000,
  "first_command": false,
  "modelflag": 7,
  "repeat": true,
  "id_restaurants": [42, 43],
  "enable_hours": null,
  "customers_ids": [142, 143]
}
```

## Réponse

```
{
  "errno": 0
}
```

Uniquement création d'une promotion. La mise à jour d'une promotion existante n'est pas possible.

#### BODY

Nom	Type	Requis	Description	Exemple
name	string	<b>Yes</b>	Nom de la promotion	<i>Promotion Test</i>
promo_type	string	<b>Yes</b>	Type de promotion : - <code>total</code> : remise globale Seul <code>total</code> est possible.	<i>total</i>
fixed_discount	int	<b>Yes</b>	Remise fixe en € Exclusif avec <code>percent_discount</code> Ce champs ne peut pas être <code>null</code> si <code>percent_discount</code> l'est aussi	<i>500</i>
percent_discount	int	<b>Yes</b>	Remise en % Exclusif avec <code>fixed_discount</code> Ce champs ne peut pas être <code>null</code> si <code>fixed_discount</code> l'est aussi	<i>1000</i>
valid_from	datetime		Valide à partir de	<i>2022-01-01 10:30:00</i>
valid_until	datetime		Valide jusqu'à	<i>2022-12-31 10:30:00</i>
min_price	int		Montant minimum du panier requis. Exprimé en cents	<i>1500</i>
disable	bool		Activation/désactivation de la promotion	<i>false</i>
noweb	bool		La promotion n'est pas valable sur le web	<i>true</i>
nopos	bool		La promotion n'est pas valable en caisse	<i>false</i>
ignore_order_min_price	bool		Quand cette promotion est appliquée, ne pas tenir compte du minimum de commande habituel	<i>false</i>
force_web_payment	bool		Forcer le paiement en ligne sur le web	<i>false</i>
only_once	bool		Promotion valable une seule fois par client	<i>true</i>
customer_registered_days	int		Valable uniquement pour les clients inscrits depuis moins de X jour(s)	<i>10</i>
max_uses	int		Nombre maximum d'utilisation de la promotion <code>null</code> = la promo peut être utilisée sans limite	<i>1000</i>
first_command	bool		Promotion valable uniquement pour la première commande	<i>false</i>
Modes dans lesquels la promotion est valable				

modelflag	int	<i>1 = sur place</i> <i>2 = à emporter</i> <i>4 = livraison</i> <i>1 + 2 + 4 = tous les 3 modes</i>	7
repeat	bool	Si la promotion doit être répétée autant que possible (ex : un cookie offert pour 5€ de commande => 3 cookies pour 15€)	true
id_restaurants	int[]	Liste des ID Zelty des restaurants éligibles <i>Si vide, alors la promotion est valable sur l'ensemble des restaurants du réseau</i>	[42, 43]
enable_hours	Timetable	Créneaux horaires de validité de la commande <code>null</code> = pas de limite	null
customers_ids	int[]	Liste des ID Zelty des clients éligibles, à associer à cette promotion <i>Si vide, alors la promotion est valable sur l'ensemble des clients du réseau</i>	[142, 143]

## POST /canUseCoupon

### Exemple de requête

```
curl https://api.zelty.fr/2.9/canUseCoupon \
-H 'Authorization: Bearer <token_api>'
```

### Body

```
{
  "id_coupon": 42,
  "customer_id": 44
}
```

### Réponse

```
{
  "ret": 0,
  "errno": 0
}
```

Permet de vérifier qu'un coupon soit encore valide

#### BODY

Nom	Type	Requis	Description	Exemple
id_coupon	int	Yes	ID Zelty de la promotion	42
customer_id	int	Yes	ID Zelty de du client	44



RÉPONSE

Si le retour est : `"ret": 0`, alors la réponse est **OK**.  
Les autres retours sont des erreurs.

Entité: Promotion

Exemple

```
{
}
```

`Promotion` représente une promotion.

ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int	Yes	ID Zelty (read-only)	42
code	string		Code promo S'il n'y a pas de code promo alors la promotion est visible automatiquement	ZeltyPromo
name	string	Yes	Nom de la promotion	Super Promotion
valid_from	datetime		Valide à partir de Date au format ISO 8601	2017-04- 28T19:18:49+00:00
valid_until	datetime		Valide jusqu'à Date au format ISO 8601	2017-04- 28T19:18:49+00:00
min_price	int		Montant minimum du panier requis. Exprimé en cents	1500
promo_type	string		Type de promotion : - <code>total</code> : remise globale - <code>items</code> : produits offerts/remisés - <code>least_expensive</code> : produit le moins cher offert/remisé parmi une liste - <code>same_item</code> : produit identique offert/remisé parmi une liste	total
required_items	Item[][]		Liste des produits requis dans le panier avant d'appliquer la promotion	Voir <b>Entité: Item</b>
only_once	bool		Promotion valable une seule fois	true

		par client	
first_command	bool	Promotion valable uniquement pour la première commande	false
max_uses	int	Nombre maximum d'utilisation de la promotion <i>null = la promo peut être utilisée sans limite</i>	1000
modeflag	int	Modes dans lesquels la promotion est valable <i>1 = sur place</i> <i>2 = à emporter</i> <i>4 = livraison</i> <i>1 + 2 + 4 = tous les 3 modes</i>	7
noweb	bool	La promotion n'est pas valable sur le web	true
nopos	bool	La promotion n'est pas valable en caisse	false
enable_hours	Timetable	Créneaux horaires de validité de la commande <i>null = pas de limite</i>	null
repeat	bool	Si la promotion doit être répétée autant que possible (ex : un cookie offert pour 5€ de commande => 3 cookies pour 15€)	true
priority	int	Priorité à laquelle appliquer la promotion <i>Ne concerne que les promotions sans code</i>	10
disable	bool	Activation/désactivation de la promotion	false
ignore_order_min_price	bool	Quand cette promotion est appliquée, ne pas tenir compte du minimum de commande habituel	false
force_web_payment	bool	Forcer le paiement en ligne sur le web	false
customer_registered_days	int	Valable uniquement pour les clients inscrits depuis moins de X jour(s)	10
customers_ids	int[]	Liste des ID Zelty des clients éligibles	[142, 143]
		Liste des ID Zelty des restaurants éligibles	

id_restaurants	int[]	Si absent, alors la promotion est valable sur l'ensemble des restaurants du réseau	[42, 43]
uses	int	Nombre de fois que la promotion a été utilisée	5
fixed_discount	int	Remise fixe en € Exclusif avec percent_discount	500
percent_discount	int	Remise en % Exclusif avec fixed_discount	10000
restaurants_validity_dates	RestaurantsValidity[]	Liste des dates d'activation par restaurants	Voir <b>Entité: RestaurantsValidity</b>
promo_data	object	Dépend de promo_type	Voir <b>Entité: Promo_data</b>

## Entité: RestaurantsValidity

### Exemple

```
{
  "id_restaurant": 42,
  "restaurant_remote_id": "m43",
  "valid_from": "2017-04-28T19:18:49+00:00",
  "valid_until": "2017-04-28T19:18:49+00:00"
}
```

`RestaurantsValidity` représente les dates d'activation dans le restaurant

### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id_restaurant	int	Yes	ID Zelty du restaurant	42
restaurant_remote_id	string		Identifiant distant du restaurant	m43
valid_from	datetime		Valide à partir de Date au format ISO 8601	2017-04-28T19:18:49+00:00
valid_until	datetime		Valide jusqu'à Date au format ISO 8601	2017-04-28T19:18:49+00:00

## Entité: Item

### Exemple

```
{
  "id": 42,
  "type": "dish",
  "remote_id": "m43"
}
```

**Item** représente un produit ou un menu.

#### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
id	int	<b>Yes</b>	ID Zelty	42
type	string	<b>Yes</b>	Type de produit	dish/menu
remote_id	string		Identifiant distant	m43

## Entité: promo\_data

### Exemple

Pour la valeur "items" la valeur de "items\_groups" est une liste de groupes d'Items.

Pour chaque groupe d'items, un produit pour être choisi par le client.

Par exemple pour offrir une boisson au choix + un dessert au choix "items\_groups" contiendra une liste de boisson + une liste de desserts.

```
[
  // Liste des boissons
  [
    // Coca-Cola
    {
      "id": 21,
      "type": "dish",
      "remote_id": "SB21"
    },
    // 7up
    {
      "id": 22,
      "type": "dish",
      "remote_id": "SB22"
    }
  ],
  [
    // Cookie
    {
      "id": 23,
      "type": "dish",
      "remote_id": "SB23"
    },
    // Glace vanille
  ]
]
```

```
{
  "id": 24,
  "type": "dish",
  "remote_id": "SB24"
}
```

De cette manière le client pourra choisir deux produits :

- Un coca ou un 7up
- Et un cookie ou une glace vanille

Le champs `promo_data` décrit la promotion en fonction du champs “`promo_type`”.

ATTRIBUTS

promo_type	Description	promo_data		
total	Remise globale. <code>percent_discount</code> ou <code>fixed_discount</code> s'applique au total de la commande		null	
items	Produits remisés. <code>percent_discount</code> ou <code>fixed_discount</code> s'applique sur chaque produit lié à la promotion	items_groups	Item[][]	Combinaisons des produits à ajouter
least_expensive	Produit le moins cher remisé parmi une liste. <code>percent_discount</code> ou <code>fixed_discount</code> s'applique sur chaque produit lié à la promotion	qty	int	Nombre de produits à commander pour appliquer la remise
		items	Item[]	Liste des produits éligibles
same_item	Produit identique remisé parmi une liste. <code>percent_discount</code> ou <code>fixed_discount</code> s'applique sur chaque produit lié à la promotion	qty	int	Nombre de produits à commander avant d'ajouter le produit remisé
		items	Item[]	Liste des produits éligibles

# Les appels téléphoniques

## POST /phonecalls

Exemple de requête

```
curl https://api.zelty.fr/2.9/phonecalls \
```

```
-H 'Authorization: Bearer <token_api>'
```

### Body

```
{
  "remote_id": null,
  "received_at": "2021-01-01T07:00:00+02:00",
  "ended_at": "2021-01-01T07:00:00+02:00",
  "status": "ringing",
  "phone_number": "+33612345678"
}
```

### Réponse

```
{
  "errno": 0
}
```

Création d'un appel téléphonique.

Cet appel doit se faire au niveau d'un restaurant.

Avec une clé API enseigne, il faudra renseigner "id\_restaurant".

#### BODY

Le corps de la requête doit contenir un objet de type `Phonecall`.

#### RÉPONSE

N/A

## Entité: Phonecall

### Exemple

```
{
  "remote_id": null,
  "received_at": "2021-01-01T07:00:00+02:00",
  "ended_at": "2021-01-01T07:00:00+02:00",
  "status": "ringing",
  "phone_number": "+33612345678"
}
```

`Phonecall` représente une appel téléphonique.

#### ATTRIBUTS

Nom	Type	Requis	Description	Exemple
remote_id	string	ID externe		m42
		Date de début de l'appel. Si ce champs est vide, la date/heure actuelle sera		2017-04-

received_at	datetime		<i>appliquée</i> <i>Date au format ISO 8601</i>	28T19:18:49+00:00
ended_at	datetime		Date de fin de l'appel <i>Si ce champs est vide, la date/heure actuelle sera appliquée</i> <i>Date au format ISO 8601</i>	2017-04-28T19:18:49+00:00
status	string	Yes	Statut de l'appel <i>ringing</i> <i>aborted</i> <i>in_progress</i> <i>ended</i>	<i>in_progress</i>
phone_number	string	Yes	Numéro de téléphone. <i>Format international E164</i>	+33612345678

## Les webhooks

### Introduction aux Webhooks

Les webhooks sont des "fonctions de rappels HTTP définis par l'utilisateur". Ils sont généralement déclenchés par un événement, comme la mise à jour d'une donnée...

Lorsque cet événement se produit, le site source envoie une requête HTTP à l'URL configurée pour le webhook.

Nos Webhooks enverrons des informations en JSON.

### Liste des Webhooks

Webhook	Description/Déclencheur	Evenement	Entité liée
Les restaurants	Récupération d'un restaurant qui vient d'être mis à jour ou qui vient d'être créé.	restaurant.update restaurant.delivery_zones.update	Voir <b>Entité: Restaurant</b>
Les produits	Récupération d'un produit qui vient d'être mis à jour ou qui vient d'être créé.	dish.update dish.delete	Voir <b>Entité: Dish</b>
Les menus	Récupération d'un menu qui vient d'être mis à jour ou qui vient d'être créé.	menu.update menu.delete	Voir <b>Entité: Menu</b>
Les options	Récupération d'une option qui vient d'être mis à jour ou qui vient d'être créée.	option.update	Voir <b>Entité: Option</b>
Les tags	Récupération d'un tag qui vient d'être mis à jour ou qui vient d'être créé.	tag.update tag.delete	Voir <b>Entité: Tag</b>

Modification d'un produit	Récupération des modifications faites sur un produit qui vient d'être mis à jour <b>au niveau restaurant seulement.</b>	dish_override.update	Voir <b>Entité: DishOverride</b>
Modification d'un menu	Récupération des modifications faites sur un menu qui vient d'être mis à jour <b>au niveau restaurant seulement.</b>	menu_override.update	Voir <b>Entité: MenuOverride</b>
Les clients	Récupération d'une fiche client qui vient d'être mise à jour ou qui vient d'être créée.	customer.update customer.delete	Voir <b>Entité: Customer</b>
Disponibilité des options	Récupération de l'état de disponibilité d'une option qui vient d'être mis à jour <b>sur la caisse seulement.</b>	option_value.availability_update	Voir <b>Entité: OptionValueAvailability</b>
Disponibilité des produits	Récupération de l'état de disponibilité d'un produit qui vient d'être mis à jour <b>sur la caisse seulement.</b>	dish.availability_update	Voir <b>Entité: DishAvailability</b>
Disponibilité des menus	Récupération de l'état de disponibilité d'un menu qui vient d'être mis à jour <b>sur la caisse seulement.</b>	menu.availability_update	Voir <b>Entité: MenuAvailability</b>
Statut des commandes	Récupération d'une commande dont le statut change en : - Production ( <b>production</b> ) - Cloturée ( <b>ended</b> ) - Annulée ( <b>cancelled</b> ) - Prête ( <b>ready</b> )	order.status.update	Voir <b>Entité: OrderStatus</b>
Cloture d'une commande	Récupération d'une commande qui vient d'être clôturée.	order.ended	Le format récupéré ressemble à la réponse de <b>GET /orders</b> mais certaines valeurs peuvent changer ( <b>price</b> par exemple)
Les clôtures de caisse	Récupération du Z de Caisse qui vient d'être effectué.	till.close	Voir <b>Entité: Closure</b>
Les réservations	Récupération d'une réservation qui vient d'être mise à jour ou qui vient d'être créée.	booking.update	Voir <b>Entité: Booking</b>
Les catalogues	Récupération d'un catalogue qui vient d'être envoyé via le back office Zelty	catalog.push	Voir <b>Entité: Catalog</b>
Les promotions	Récupération d'une promotion qui vient d'être créée ou mise à jour via le back office Zelty	promotion.update	Voir <b>Entité: Promotion</b>

## Format des Webhooks



## Exemple de webhook

```
{
  "event_id": "64807b77-d9de-4b8e-9ca9-a9eed0fcdfa3",
  "event_name": "order.status.update",
  "created_at": "2021-09-30T15:12:59.211494+02:00",
  "version": "v2",
  "brand_id": 42,
  "restaurant_id": null,
  "data": {
    "id": 62733555,
    "uuid": "000033-25bb2362-00003a-e6d41e",
    "remote_id": null,
    "status": "production"
  }
}
```

Nom	Type	Description	Exemple
event_id	string	ID unique de l'évènement	<i>64807b77-d9de-4b8e-9ca9-a9eed0fcdfa3</i>
event_name	string	Nom de l'évènement	<i>order.update</i>
created_at	datetime	Date de création de l'évènement	<i>2021-09-30T15:12:59.211494+02:00</i>
version	string	Version du webhook	<i>v2</i>
brand_id	int	ID de l'enseigne	<i>42</i>
restaurant_id	int	ID du restaurant	<i>null</i>
data	{}	Contenu du webhook	

## Signature des webhooks

### Header

```
X-Zelty-Hmac-Sha256: 1cb941a3cd10d2f68254287380901ada89c15a258f296cef2270a19e9eebb830
```

### Exemple en PHP

```
$webhook_secret = "your_webhook_secret";
$data = "our_payload";

$digest = hash_hmac('sha256', $data, $webhook_secret);
```

### Exemple en Ruby

```
require "openssl"
```

```
webhook_secret = "your_webhook_secret"
data = "our_payload"

digest = OpenSSL::Digest.new('sha256')
OpenSSL::HMAC.hexdigest(digest, webhook_secret, data)
```

Lors de l'enregistrement des webhooks (ou du compte partenaire sur le Marketplace) nous vous fournissons une clé secrète. À partir de la version 2 des webhooks, nous utilisons cette clé secrète pour signer chaque webhook. Lorsque vous recevez une requête POST, vous pourrez recalculer la signature, à partir de la clé secrète et du contenu de la requête pour vérifier l'authenticité du webhook. La signature se situe dans les headers de la requête.

## GET /webhooks

### Exemple de requête

```
curl https://api.zelty.fr/2.9/webhooks \
-H 'Authorization: Bearer <token_api>'
```

### Réponse

```
{
  "errno": 0,
  "secret_key": "*****1234",
  "webhooks": {
    "menu.update": {
      "target": "https://...",
      "created_at": "2020-11-05T14:02:02+01:00",
      "version": "v2"
    }
  }
}
```

### Récupération des webhooks

#### RÉPONSE

Nom	Type	Description	Exemple
secret_key	string	Webhook Secret	"*****1234"
webhooks	<code>object&lt;string, Webhook&gt;</code>	Objet clé/valeur dont les clés sont les noms des évènements et les valeurs la description des webhooks.	Voir <b>Entité: Webhook</b>

## POST /webhooks

### Exemple de requête

```
curl https://api.zelty.fr/2.9/webhooks \
-H 'Authorization: Bearer <token_api>'
```

Body

```
{
  "secret_key": "*****1234",
  "webhooks": {
    "menu.update": {
      "target": "https://...",
      "version": "v2"
    },
    "menu.delete": null
  }
}
```

Réponse

```
{
  "errno": 0
}
```

Création/modification d’un (ou plusieurs) webhook(s)

BODY

Le corps de la requête doit contenir un objet de type `Webhook` ou une liste d’objets de type `Webhook`.  
Les webhooks disposant d’un `event.name` existants seront mis à jours, les autres créés.  
Les `event.name` configurés à `null` seront supprimés.

Nom	Type	Requis	Description	Exemple
secret_key	string		Clé secrète. Si renseignée lors d'un POST, elle sera mise à jour.	"abcde1234"
webhooks	<code>object&lt;string, Webhook&gt;</code>	Yes	Objet clé/valeur dont les clés sont les noms des évènements et les valeurs la description des webhooks.	Voir <b>Entité: Webhook</b>

RÉPONSE

N/A

Entité: Webhook

Exemple

```
{
  {event.name}: {
    "target": "https://...",
    "created_at": "2020-11-05T14:02:02+01:00",
    "version": "v2"
  }
}
```

```
}  
}
```

`Webhook` représente un webhook.

ATTRIBUTS

Nom	Type	Requis	Description	Exemple
event.name	string	Yes	Nom de l'événement	Voir <b>Liste des webhooks</b>
target	string	Yes	URL d'ancrage	<code>https://...</code>
created_at	datetime		Date de la création du webhook, généré automatiquement	<code>2020-11-05T14:02:02+01:00</code>
version	string	Yes	Version du webhook	<code>v2</code>

# Autres

## GET /transaction-methods

Exemple de requête

```
curl https://api.zelty.fr/2.9/transaction-methods \  
-H 'Authorization: Bearer <token_api>'
```

Réponse

```
{  
  "transaction_methods": [  
    {  
      "id": 42,  
      "name": "Espèces",  
      "ico": "euros",  
      "type": "cash",  
      "order": 1,  
      "overflow": "change",  
      "allow_negative_transacion": true,  
      "open_drawer": false,  
      "disable": false,  
      "accounting_code": null  
    }  
  ]  
}
```

Récupération de la liste des méthodes de transaction

PARAMÈTRES GET

N/A

RÉPONSE

Nom	Type	Description	Exemple
id	int	ID Zelty	42
name	string	Nom de la méthode de transaction	Espèce
ico	string	Icône affichée sur la caisse	euros
type	string	Défini le type de la méthode de transaction <i>cash, ticket restaurant...</i>	cash
order	int	Défini l'ordre d'affichage	1
overflow	string	change / allow / disallow	change
allow_negative_transacion	bool	Autoriser les transactions négatives	true
open_drawer	bool	Ouverture du tiroir	false
disable	bool	Disponibilité de la méthode de transaction sur la caisse	false
accounting_code	string	Code comptable	null

## GET /transaction-methods/:id

Exemple de requête

```
curl https://api.zelty.fr/2.9/transaction-methods/<id_transaction_method> \
-H 'Authorization: Bearer <token_api>'
```

Réponse

```
{
  "transaction_method": {
    "id": 42,
    "name": "Espèces",
    "ico": "euros",
    "type": "cash",
    "order": 1,
    "overflow": "change",
    "allow_negative_transacion": true,
    "open_drawer": false,
    "disable": false,
    "accounting_code": null
  }
}
```

Récupération d'une méthode de transaction

PARAMÈTRES URL

Nom	Type	Description	Exemple
id	int	Identifiant de la méthode de transaction	42

## PARAMÈTRES GET

N/A

## GET /devices

## Exemple de requête

```
curl https://api.zelty.fr/2.9/devices \
-H 'Authorization: Bearer <token_api>'
```

## Réponse

```
{
  "devices": [
    {
      "id": 42,
      "name": "Ipad de Zelty",
      "ref": "A",
      "app_ver": "4.10",
      "app_build": "1234",
      "created_at": "2019-09-04T21:52:45+02:00",
      "last_activity": "2019-08-01T07:00:00+02:00"
    }
  ]
}
```

Récupération de la liste des appareils connectés utilisant la caisse Zelty au sein d'un restaurant

## PARAMÈTRES GET

N/A

## RÉPONSE

Nom	Type	Description	Exemple
id	int	ID Zelty	42
name	string	Nom de l'appareil connecté	<i>Ipad de Zelty</i>
ref	string	Référence Zelty	A
app_ver	string	Version de l'application Zelty Caisse	4.10
app_build	string	Build de l'application Zelty Caisse	1234
created_at	datetime	première connection de l'appareil	2019-08-01T07:00:00+02:00

last_activity	datetime	Dernière activité de l'appareil	2019-08-01T07:00:00+02:00
---------------	----------	---------------------------------	---------------------------

---

# Changelog

---

## Version 2.9

### Nouveautés :

- **BREAKING CHANGE:** Le domaine `bo.zelty.fr/app_api/` n'est plus supporté.
  - Seul le domaine `api.zelty.fr/` fonctionne à partir de cette version.

---

## Version 2.8

### 2024-08-12 :

- `GET /catalog/dishes_overrides`
  - ajout du paramètre `show_all`.
- `GET /catalog/options_overrides`
  - ajout du paramètre `show_all`.
- `GET /catalog/menus_overrides`
  - ajout du paramètre `show_all`.

### 2024-07-30 :

- Entité `OrderTransaction`
  - ajout du champs `id_transaction_method` retournant ainsi l'ID de la méthode de paiement en plus de son nom.

### 2024-07-08 :

- `GET catalog/options_overrides`
  - ajout d'un nouveau endpoint permettant de récupérer les surcharges d'options des restaurants depuis l'API.
- `POST orders/:id/transactions`
  - modification du comportement : il n'est plus possible d'envoyer des transactions à 0.

---

## Version 2.7

- `GET /catalog/dishes`
  - ajout du paramètre `limit`, par défaut 2500 `dishes` sont récupérés.
  - ajout du paramètre `offset` dans la documentation.
- `POST /catalog/dishes`
  - `options` une limite de 20 options par produits a été ajoutée.
- `POST /coupons`
  - `customers_ids` peut maintenant être utilisé pour associer des clients à une promotion lors de sa création.
- `GET /catalog/tags`
  - ajout du paramètre `all_restaurants` pour récupérer tous les tags des restaurants depuis une clé enseigne.

- `POST orders/:id/closure`
  - ajout d'un nouveau endpoint permettant de clôturer une commande depuis l'API.
- `POST orders/:id/production`
  - ajout d'un nouveau endpoint permettant d'envoyer en production des produits depuis l'API.
- Liste des Webhooks
  - ajout du webhook `promotion.update`.
- `POST /customers`
  - `check_existing` parameter now checks phone unicity
- `GET /webhooks`
  - Récupère la liste des webhooks définis
- `POST /webhooks`
  - Permet d'ajouter/modifier/supprimer ses endpoints
  - Permet également de mettre à jour le `secret_key` associé aux webhooks
  - Event catalog.push ajouté
- `GET /customers`
  - **BREAKING CHANGE:** suppression du paramètre `since`
  - **BREAKING CHANGE:** ajout des paramètres `updated_before` et `updated_after`
- `PATCH /orders/:id`
  - Calcul et sauvegarde des taxes après ajout de produits
- `GET /closures`
  - Nouveau paramètre d'url `restaurant_ids` pour filtrer sur plusieurs restaurants
- `GET /catalog/*`
  - Nouveau paramètre d'url `lang` pour télécharger la carte dans la langue souhaitée
- `GET /closures/:id/orders`
  - **BREAKING CHANGE:** ce endpoint a été supprimé
  - `GET /orders?till_id=XXX` est disponible à la place et est paginé
- `Order` entity (`GET`)
  - **BREAKING CHANGE:** La propriété `status` est désormais une `string` avec les valeurs suivantes : `opened`, `closed`, `cancelled`
  - **BREAKING CHANGE:** La propriété `mode` est désormais une `string` avec les valeurs suivantes : `eat_in`, `takeaway`, `delivery`
  - **BREAKING CHANGE:** La propriété `source` est désormais une `string` avec les valeurs suivantes : `pos`, `remote`, `web`, `ubereats`...
  - **BREAKING CHANGE:** La propriété `covers` a été renommée en `seats` (int)
  - **BREAKING CHANGE:** La propriété `ref` est désormais une `string`
  - **BREAKING CHANGE:** La propriété `price` est désormais un objet avec les valeurs suivantes:
    - `final_amount_inc_tax` (int) montant TTC de la commande
    - `final_amount_exc_tax` (int) montant HT de la commande (dispo après clôture de la commande)
    - `discounts` (array) liste des remises globales (hors remises produit). (dispo uniquement avec `expand[]=price.details`)
    - `taxes` (array) liste des taxes (dispo après clôture de la commande). (dispo uniquement avec `expand[]=price.details`)
  - **BREAKING CHANGE:** Les propriétés `total_discount` et `discount_justification` ont été supprimées



- (désormais dispo dans `price.discounts`)
- **BREAKING CHANGE:** Les propriétés `loyalty_discount` et `loyalty_used` ont été supprimées (désormais dispo dans `price.discounts`)
- **BREAKING CHANGE:** Les propriétés `coupon_id` et `coupon_code` ont été supprimées (désormais dispo dans `price.discounts`)
- **BREAKING CHANGE:** La propriété `taxes` a été supprimée (désormais dispo dans `price.taxes`)
- **BREAKING CHANGE:** La propriété `contents` a été renommée en `items` avec les changements suivants dans les items :
  - `item_id` est désormais une `string`
  - `price` est un objet (toujours présent) avec les propriétés suivantes :
    - `base_original_amount_inc_tax` (int) : prix de la ligne (hors remise et hors suppléments/options)
    - `original_amount_inc_tax` (int) : prix de la ligne avec suppléments/options
    - `discounted_amount_inc_tax` (int) : prix de la ligne avec suppléments/options et potentielle remise
    - `final_amount_inc_tax` (int) : prix final de la ligne incluant aussi les remises globales de la commande (dispo uniquement après clôture)
    - `tax` (objet) : avec les propriétés suivantes :
      - `tax_amount` (int) : montant de la TVA (dispo uniquement après clôture)
      - `tax_rate` (int|null) : taux de TVA dans le cas d'un taux unique. Dans le cas d'un menu avec plusieurs taux cette valeur sera nulle
  - `contents` est renommé en `modifiers` avec un nouveau format (voir la doc)
  - **BREAKING CHANGE:** La propriété `items` n'est plus retournée automatiquement. Il faut passer `expand[]=items`
- **POST /orders**
  - **BREAKING CHANGE:** Les propriétés `items[].options` et `items[].dishes[].options` ont été supprimées
  - Nouvelles propriétés `items[].modifiers` et `items[].dishes[].modifiers`. Array d'objets avec le format suivant:
  - `option_id` (int) : ID du groupe d'options
  - `option_value_id` (int) : ID de la valeur
  - `quantity` (int) : quantité de l'option (vide = 1)
  - `price` (int) : prix de l'option (vide = 0)
  - **BREAKING CHANGE:** La propriété `mode` est désormais une `string` avec les valeurs suivantes : `eat_in`, `takeaway`, `delivery`
- **GET /orders**
  - Ajout d'un filtre (optionnel) `mode` dans l'URL. Les valeurs possibles sont : `mode=eat_in`, `mode=takeaway`, `mode=delivery`
- **GET /fabrication-places**
  - Ce endpoint a été créé
- **GET /transaction-methods**
  - Ce endpoint a été créé, l'endpoint `GET catalog/transaction_methods` reste toujours disponible, cependant, à terme, il sera supprimé pour laisser place à `GET /transaction-methods` uniquement

## Version 2.6

- **Option** entity
  - **BREAKING CHANGE:** La propriété `nb` est remplacée par `max_choices`
  - **BREAKING CHANGE:** La propriété `mandatory` est remplacée par `min_choices` (int)

## Version 2.5

- `POST /orders/:id/transactions`
  - Ajout de la propriété `remote_id` (uuid, optional) dans chaque transaction pour rendre la requête d'ajout d'une transaction idempotente
- `Order` entity:
  - **BREAKING CHANGE:** La propriété `taxes` a été supprimée
  - **BREAKING CHANGE:** La propriété `price` va être modifiée par un objet avec les valeurs suivantes :
    - `amount_inc_tax` => le prix ttc "final" du produit, c'est à dire calculé après toutes remises (produit ou globale)
    - `tax_amount` => le montant de TVA sur cette ligne
    - `tax_rate` => le taux de TVA uniquement si l'ensemble des produits à l'intérieur est soumis au même taux de TVA (sinon ça sera `null`)
  - **BREAKING CHANGE:** Dans les menus :
  - Ajout d'une nouvelle propriété `price` dans chaque ligne du menu avec les mêmes propriétés que ci-dessous.
  - **BREAKING CHANGE:** La valeur `contents.taxes` du champ `expand` est remplacée par `contents.price`

## Version 2.4

- `POST /orders`
  - Nouvelle propriété `display_id`
  - Nouvelle propriété `virtual_brand_name`
  - **BREAKING CHANGE:** Propriété `places` renommée en `seats`
  - **BREAKING CHANGE:** Suppression de `is_mobile` et `is_kiosk` qui sont remplacée par `source`
- Nouveau paramètre `expand[]` disponible pour `GET /orders`, `GET /orders/:id`, `GET /closures/:id/orders`
  - Cette propriété permet de choisir la liste des entités que l'on souhaite récupérer dans l'objet `Order`
  - **BREAKING CHANGE:** Par défaut plus aucune entité n'est retournée si on ne passe pas la propriété `expand`

## Version 2.3

- `POST /orders` : la zone de livraison est désormais optionnelle
- Nouvelle propriété `meta` disponible dans la liste des zones de livraisons
- Nouveau paramètre `fields[]` disponible sur `GET /restaurants`
  - Permet de choisir les entités que l'on souhaite retourner dans un restaurant `billing` et `delivery_zones`
  - **BREAKING CHANGE:** sans ce paramètre les propriétés `billing` et `delivery_zones` ne sont plus retournées

## Version 2.2

- Nouveaux paramètres `opened=1` et `is_sandbox=1` dans `GET /orders`

- Possibilité de modifier un commentaire `PATCH /orders {"comment": "toto"}`
- Nouveau paramètre `all_restaurants` dans `GET /catalog/options`
- **BREAKING CHANGE:** Nouveau paramètre `show_all` (par défaut = false) `GET /catalog/options`
- **BREAKING CHANGE:** renommage de la propriété `tags` en `taxes` dans `GET /catalog/taxes`

## Version 2.1

- **BREAKING CHANGE:** `POST /orders` une option introuvable retourne désormais un erreur
- **BREAKING CHANGE:** `GET /orders`, `GET /orders/:id` et `GET /closures/:id/orders` : harmonisation de l'objet Orders, plusieurs propriétés ont été renommées/supprimées/ajoutées

# Les erreurs

## Entité: Liste des erreurs

Pour chaque appel est retourné un objet JSON contenant les paramètres suivants :

### ATTRIBUTS

Nom	Type	Description	Exemple
errno	int	Numéro de l'erreur 0 = pas d'erreur	5
errmsg	string	La description de l'erreur errno seulement si errno != 0	La clef d'API est incorrecte

## Entité: Liste des valeurs errno

### ATTRIBUTS

Numéro	Code HTTP	Message	Commentaire
0	200	Pas d'erreur	
-2	400	Paramètre invalide	Paramètre entrant inattendu (get ou post)
-1	500	Erreur serveur	
2	404	Endpoint introuvable	Couple "METHOD/path" inexistant
4	400	Version d'API inconnue	

5	403	Clef d'API incorrecte ou révoquée	
65	400	Date invalide	
70	400	JSON Invalide	
71	400	Erreurs de validation	<i>Voir le champ "errors" qui détaille les erreurs de validation</i>
404	404	Élément introuvable	<i>Produit/menu/restaurant introuvable</i>