

# BDE ESEOmega

## Dossier de passation

© *SheepDev* 2015-2016



# Chapter 1

## Généralités

### 1.1 API

#### 1.1.1 Base

L'API développée communique avec les différents services par le biais des interfaces PHP qui seront énumérés par la suite.

Le serveur est accessible à l'adresse `https://web59.secure-secure.co.uk/francoisle.fr/`, ainsi qu'en utilisant l'adresse dépendant de l'IP, `http://217.199.187.59/francoisle.fr/`.

Préférer l'adresse HTTPS, afin de renforcer la sécurité des transactions (rendant l'interception par Wireshark impossible).

Le chemin de base de l'API (racine) est celui du serveur (`https / IP`) suivi de `lacommande/api/`. Dans la suite de ce document, toutes les interfaces seront référencées vis à vis de ce chemin.

#### 1.1.2 Paramètres

Pour passer des paramètres aux interfaces PHP du serveur, il faut utiliser la méthode d'accès POST, définie dans le protocole HTTP.

#### 1.1.3 Réponse

Les réponses des interfaces du serveur respectent le format de données JSON. C'est plus beau que le XML, plus flexible que le CSV, et bien plus connu que le YAML.

Chaque réponse du serveur contient **systématiquement** les données suivantes :

- **status** : Code d'erreur (1 : l'opération a réussi, sinon en cas d'erreur :  $-x$  avec  $x$  étant un nombre)
- **cause** : Message associé à ce code d'erreur. Peut être vide (lorsque **status** est à 1 entre autres)
- **data** : Objet contenant les données liées à la requête vers l'API. Peut être vide.<sup>1</sup>

---

<sup>1</sup>Attention, lorsque **data** est vide, le champ est présent dans le JSON comme étant un *Array* et non un *Object* (et uniquement dans cette situation-là). Sous Android, préférer la fonction `optJSONObject(...)` au lieu de `getJSONObject(...)`, sinon ça guy-plantera.

### 1.1.4 Exemples

#### Sans paramètres requis

Pour obtenir le message de service de la cafétéria, la requête est faite sur l'interface `info/service.php`.

Etant donné qu'il ne s'agit que d'une information, aucun paramètre POST n'est requis. La réponse du serveur est la suivante (ça reste un exemple hein) :

```
{
  "status":1,
  "cause": "",
  "data":{"
    "service":"Le BDE paie sa danse du crabe"
  }
}
```

#### Avec paramètres manquants

La grande majorité des interfaces ont besoin de paramètres POST pour savoir quoi faire et avec qui. Si ces paramètres sont manquants (il en suffit d'un !), ou qu'une valeur ne correspond pas au formalisme attendu, une erreur est générée, et la réponse ressemblera, en gros, à ça :

```
{
  "status":-1,
  "cause":"Parametres invalides",
  "data":[ ]
}
```

Au passage, vous noterez ici le bel exemple du champ `data` qui, une fois vide, se transforme en *Array*, ah bah bravo Morray.

#### Avec paramètres invalides

Tentative de récupération du numéro de version à jour d'une application, depuis l'interface PHP `info/version.php`. Pas besoin d'en dire plus (enfin d'expliquer pourquoi ça marche pas).

Paramètres :

`os = DICK-HEAD`

Réponse :

```
{
  "status":-2,
  "cause":"\"DICK-HEAD\" n'est pas declaree comme etant une application",
  "data":[ ]
}
```