

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 (sauf exception, comme pour les news, sponsors, clubs, etc.).

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.¹

¹Attention, 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":[ ]
}
```

Chapter 2

News

Les applications ESEOmega, ainsi que le site internet du BDE, peuvent récupérer les dernières news publiées par les clubs (ainsi que par le BDE lui même). L'accès à cette interface de l'API se fait en utilisant l'URL suivante :

`eseonews/getData.php` (**Attention !** référence par rapport au sous serveur `francoisle.fr`)

2.1 Paramètres

L'interface ne requiert aucune donnée type authentification (les news ne sont pas un contenu protégé). En revanche, deux paramètres `GET`¹ sont requis :

- `height` : le nombre de news que l'on désire récupérer
- `ptr` : le décalage en multiple de `height` que l'on souhaite appliquer pour la récupération des news

2.1.1 Exemple

On désire afficher les 5 premières news :

`height = 5 & ptr = 0`

2.1.2 Exemple encore plus

On désire afficher les 3 news qui suivent les 3 premières news (donc les news 3 à 5, si la première news est référencée comme étant la 0):

`height = 3 & ptr = 1`

En suivant cette logique, si on voulait afficher les 3 news qui suivent encore, il suffirait de faire passer le paramètre `ptr` à 2, puis 3 ... vous avez compris.

C'est notamment ce qui est utilisé dans les applications pour obtenir un scroll *end-less*.

¹Le `POST` on garde ça pour les champs qui servent à identifier quelque chose comme un utilisateur, t'as vu



Figure 2.1: Scroll *end-less* : chargement automatique des news

2.2 Réponse

Les données retournées (dans le cas où au moins une news existe) sont les suivantes :

- **articles** : la liste des articles en tant qu'array JSON

Pour chaque article :

- **id** : l'identifiant de la news en question, la plus récente ayant le nombre le plus élevé
- **titre** : le titre de la news, sans formatage HTML
- **date** : la date de rédaction de la news, au format SQL²
- **lien** : le lien (URL) vers l'article sur le site du BDE. Peut être vide (si aucun site n'existe par exemple. Dans ce cas, l'application ne doit pas permettre le clic vers cette URL ...)
- **auteur** : le nom du club qui a rédigé la news en question
- **img** : le lien vers l'image associée à ce club (image de header)
- **content** : le contenu de la news à afficher, avec formatage HTML³
- **resume** : le résumé de la news, qui peut être utilisé pour afficher dans une liste le début du contenu (valeur sans formatage HTML)

²Pour rappel, le format SQL c'est ça : YYYY-MM-DD HH:mm:ss

³Attention, le formatage HTML utilisé est celui du WYSIWYG (éditeur graphique). Il ne contient donc pas les champs `<html>`, `<body>`, etc. Pour une webview dans une application mobile, cela ne pose pas de problèmes, en revanche, pour un site web, il conviendra de respecter les normes imposées par le W3C en ajoutant les balises qu'il faut ...

2.2.1 Exemple

On utilise l'interface de l'API news avec les paramètres suivants :

`height = 2 & ptr = 0`

Et ce, de façon à obtenir les deux dernières news les plus récentes.

```
{
  "articles": [
    {
      "id":1234,
      "titre":"SheepDev est l\u00e0 !",
      "date":"2016-05-30 20:59:14",
      "auteur":"Shep4Ever",
      "lien":"http://eseomega.fr/article-1234",
      "img":"https://web59.secure-secure.co.uk/
francoisle.fr/eseonews/jsondata/clubs_img/sheep.jpg",
      "content":"<div style=\"background-color: #fff7e5;
margin: 0; padding: 0\">Caresses sur les cheveux</div>",
      "resume":"Caresses sur les cheveux"
    },
    {
      "id":1233,
      "titre":"L'ESEO championne de Python !",
      "date":"2016-05-29 20:12:13",
      "auteur":"Administration ESEO",
      "lien":"http://eseomega.fr/article-1233",
      "img":"https://web59.secure-secure.co.uk/
francoisle.fr/eseonews/jsondata/clubs_img/rock-bert.jpg",
      "content":"<span style=\"font-size: 12px; font-family: Arial;
color: rgb(69, 69, 69); background-color: transparent;
font-variant-ligatures: normal;
font-variant-position: normal;
vertical-align: baseline;
white-space: pre-wrap;\">
L'ESEO termine en 1 ère place à son codeur
Python Axel Rollo ! Much wow</span>",
      "resume":"L'ESEO termine en 1 ère place à son codeur
Python Axel Rollo ! Much wow"
    }
  ]
}
```

Le conseil du chef : Pour un steak bien réussi, n'ajoutez du sel et du poivre qu'une fois que le côté à assaisonner est totalement cuit. Cela vous évitera d'avoir du sel ou du poivre brûlé sur le steak, ainsi que sur la poêle !

Autre conseil : n'oubliez pas que même si `height` vaut par exemple 2 lors de votre requête, vous n'aurez pas forcément 2 news dans l'array JSON de la réponse (eh oui, il se peut qu'un seul article n'ait été posté). En règle générale, n'utilisez pas `height` comme une référence pour parser et afficher vos données, mais plutôt la taille de l'array JSON de la réponse (utilisez `JSON.length()` par exemple sur Android)