

Énoncé de l'exercice :

Il s'agit de développer une petite application permettant d'accéder à l'application LMB par API.

Les fonctions attendues sont les suivantes :

- rechercher des clients
- ouvrir une fiche client
- éditer une fiche client

Objectif n°1 : **Respecter la maquette fournie**

Objectif n°2 : **Accéder à LMB par API-REST (PHP, CURL)**

Objectif n°3 : **Faire fonctionner l'appli**

Objectif n°4 : **Utiliser AngularJS pour le formulaire d'édition (sinon JQuery)**

Vous disposez pour cet exercice de toutes les ressources Internet de votre choix.

Le délai maximum de réalisation est de 1 journée.

La documentation de l'API est fournie ci-après.

Les maquettes également.

Compte rendu :

Merci de m'adresser un compte rendu par email précisant :

- Les objectifs réalisés et non réalisés
- Le temps de réalisation de chaque étape

Notes:

- Vous pouvez utiliser Google
- Vous pouvez poser des questions à nos techniciens (et c'est même encouragé)
- Vous ne devez pas utiliser de framework php (laravel, symphony...)
- L'appel API est plutôt à faire en PHP qu'en JS.

Maquettes

Rechercher un contact

Recherche d'une fiche de contact

Renseigner un nom ou une dénomination

Nom ou dénomination

Rechercher

	Nom	Adresse	Ville	Téléphone	
PM	Paul MICHEL				<div>Q Voir</div>
MB	Mathieu BOMPART	836 rue du Mas de Verchant	34000 - MONTPELLIER	01 23 45 67 89	<div>Q Voir</div>
CP	Céline PERIK	836 rue du Mas de Verchant	34000 - MONTPELLIER	09 87 65 43 21	<div>Q Voir</div>
JB	Jean-Yve BAUDIN				<div>Q Voir</div>
DR	Dominguez ROGER	836 rue du Mas de Verchant	34000 - MONTPELLIER	02 34 56 78 90	<div>Q Voir</div>

Rechercher un contact

Mathieu BOMPART

Editer

INFORMATIONS

Prénom & NOM

Téléphone

Email

Adresse

Mathieu BOMPART

01 23 45 67 89

mathieu@bompart.mb

836 rue du Mas de Verchant
34000 Montpellier

Mathieu **BOMPART**

✎ Editer

EDITION

Prénom & NOM

Mathieu BOMPART

Téléphone

0123456789

Email

mathieu@bompart.mb

Adresse

836 rue du Mas de Verchant

Code postal

34000

Ville

Montpellier

Annuler

Enregistrer

Documentation API REST

URL à interroger: URL_LMB/api

ex: <https://evaluation-technique.lundimatin.biz/api/>

Api versioning et format de retour:

- A préciser dans le accept de l'en-tête HTTP: " Accept: application/api.rest-v1+json "

POST, PUT

- Le format des données sur les POST/PUT est le JSON, le content-type est à préciser dans l'en-tête HTTP: " Content-Type: application/json "

Retour des messages:

- Le code de retour est retourné dans le Code de statut HTTP.
 - 200 : OK
 - 201 : Objet créé (sur POST de création d'objets/correspondance)
 - 400 : Paramètres d'entrées erronés/manquants
 - "Versionning erroné" => vérifiez "Accept: application/api.rest-v1+json"
 - 401 : Login/Mot de passe incorrects (auth)
 - 403 : Accès refusé (appel api avec mauvais token)
 - 404 : Appel API non trouvé
 - 405 : Droits insuffisants pour exécuter la requête demandée
 - 410 : Incompatibilité de version
 - 415 : Le format de données transmis ne match pas le content-type précisé
 - 500 : Erreur interne
- index "code" : répétition du code HTTP.
- index "message" : vide ou contient un message verbeux (message d'erreur)
- index "datas" contient le retour de la méthode appelée.
- index "warnings" : liste de messages

Codes d'erreurs :

- SYSTEM_ERROR : erreur système
 - MISSING_REQUIRED_PARAMETERS : paramètres requis manquants
 - ELEMENT_NOT_FOUND : élément non trouvé
 - INVALID_DATA_FORMAT : format des données invalide
 - PROCESS_FAILURE : erreur durant le processus
-

AUTHENTIFICATION VIA API (pour l'utilisateur connecté)

L'authentification se fait en 2 temps.
Dans un premier il faut récupérer le token d'authentification via un appel spécifique à l'api, puis passer ce token dans tous les appels API suivants.

Récupération du token d'authentification: (Authentication basic HTTP non nécessaire pour cet appel)

[POST] /auth

Liste des paramètres POST :

Nom	Type	Obligatoire	Description
username	string	OUI	Identifiant de connexion à l'API
password	string	OUI	Mot de passe en clair ou hashé selon la valeur passée à password_type
password_type	enum(0,10)	NON	0 permet de définir que le mot de passe est en clair, 10 permet de définir que le mot de passe est déjà hashé Par défaut, la valeur utilisée est 0.
code_application	enum(webs ervice_exter ne)	OUI	Permet de définir quel type d'application se connecte à l'API.
code_version	string	OUI	Permet de définir quelle version de l'application appelante est utilisée. (ex: geniusApp-v1.0)

Exemple d'appel à l'API d'authentification :

```
{
  "username": "test_api",
  "password": "api123456",
  "password_type": 0,
  "code_application": "webservice_externe",
  "code_version": "1"
}
```

Données de retour :

Nom	Type	Description
token	string	jeton d'authentification est à passer dans tous les appels ultérieurs à l'api

Exemple de retour sur un POST /auth :

```
{
  "code":200,
  "message":"",
  "datas":{
    "token":"91c86q8t86rb013epv0vgks7i5"
  },
  "warnings":[
  ]
}
```

Le token d'authentification est à passer dans tous les appels à l'api autre que l'authentification elle-même, dans le mot de passe de l'en-tête HTTP "Authorization". Le login sera toujours vide.

API CLIENTS

- GET /clients - Récupérer une liste de clients
- GET /clients/C-00003 - Récupérer un client spécifique
- PUT /clients/C-00003 - Mettre à jour le client qui a pour id : C-00003

clients - Liste des clients (Authentification HTTP requise)

[GET] /clients/

Liste des paramètres GET possibles:

Nom	Obligatoire	Description
nom	NON	Permet de filtrer sur un nom (expression incomplète)
ville	NON	Permet de filtrer sur une ville (expression incomplète)
sort	NON	Permet de trier les résultats suivant plusieurs paramètres possibles(séparer par des virgules) à savoir: <ul style="list-style-type: none">- nom (-nom = DESC, nom/+nom=ASC)- date_creation- id- ville
fields	NON	Permet de définir quel champs du client retourne l'API
limit	NON	Permet de mettre une limite sur la recherche de clients

Données de retour : tableau de clients

Nom	Description	Présence
id	Référence du client	Toujours
nom	Nom/Prénom du client	Suivant paramètre fields
date_creation	Date de création du client	Suivant paramètre fields
email	Email du client	Suivant paramètre fields
tel	Téléphone du client	Suivant paramètre fields
adresse	Adresse du client	Suivant paramètre fields
code_postal	Code postal du client	Suivant paramètre fields

ville	Ville du client	Suivant paramètre fields
-------	-----------------	--------------------------

exemple de retour: pour les paramètres suivants: sort=-nom&fields=nom,ville,code_postal&limit=2

```
{
  "code":200,
  "message":"",
  "datas":[
    {
      "id":"C-00004",
      "nom":"MIOSSEC Christophe",
      "ville":"BREST",
      "code_postal":"29200"
    },
    {
      "id":"C-00003",
      "nom":"DOMINGUEZ Roger",
      "ville":"MONTPELLIER",
      "code_postal":"34000"
    }
  ],
  "warnings":[]
}
```

clients - Récupération d'un client (Authentication HTTP requise)

[GET] /clients/{ID}
({ID} = référence d'un client)

Données de retour : un client avec les données suivantes

Nom	Description
id	Référence du client
nom	Nom/Prénom du client
email	Email du client
tel	Téléphone du client
adresse	Adresse du client
code_postal	Code postal du client
ville	Ville du client

clients - Mettre à jour un client (Authentication HTTP requise)

[PUT] /clients/{ID}
({ID} = référence d'un client)

Liste des paramètres POST possibles:

Nom	Obligatoire	Description
nom	NON	Mettre à jour le nom (si précise, doit être non vide)
email	NON	Mettre à jour l'email
tel	NON	Mettre à jour le téléphone
adresse	NON	Mettre à jour l'adresse
code_postal	NON	Mettre à jour le code postal
ville	NON	Mettre à jour la ville