TP transfert de données

May 10, 2016

1 API

1.1 Définition

> En informatique, une interface de programmation (souvent désignée par le terme API pour Application Programming Interface) est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une bibliothèque logicielle ou un service web, le plus souvent accompagnée d'une description qui spécifie comment des programmes consommateurs peuvent se servir des fonctionnalités du programme fournisseur.

source: Wikipedia

Plus simplement, une API est un ensemble de pages webs bien définies qui permettent de transférer des informations/données. Une des problèmatiques actuelles en science est le "big data". En effet, il est de plus en plus facile et rapide d'obtenir des données expérimentales. Il a donc fallu trouver des moyens de partager ces informations. Pour cela de nombreuses API ont été mises en place.

Vous utilisez, sans le savoir, de nombreuses API quotidiennement. Lorsque vous vous connectez sur un site en utilisant votre compte facebook, gmail, vous utilisez une API. Les applications mobiles qui se connectent pour récupérer des informations utilisent toutes des API (météo, GPS, transilien, facebook, etc...). Dès que vous avez de l'autocomplétion dans un formulaire, il y a une API derrière.

Vous trouverez en index une liste d'API pour exemple.

Durant ce TP nous allons voir et utiliser les différents éléments composants une API.

2 Les formats de transfert de données

Il existe différents formats de données destinés au transfert informatique d'informations. On peut citer, parmis les plus connus et les plus utilisés, le JSON, le XML ou encore le YAML. Vous trouverez ci-dessous un exemple décliné dans ces trois

formats. (Il n'est pas nécessaire de tous les comprendre). Nous utiliserons dans le tp, le JSON qui sera détaillé plus loin.

2.1 XML: Etensible Markup Language

```
<menu id="file" value="File">
 <popup>
  <menuitem value="New" onclick="CreateNewDoc()" />
  <menuitem value="Open" onclick="OpenDoc()" />
  <menuitem value="Close" onclick="CloseDoc()" />
 </popup>
</menu>
```

2.2 JSON: JavaScript Object Notation

```
{
 "menu":
  {
   "id": "file",
   "value": "File",
   "popup": {
    "menuitem": [
      { "value": "New", "onclick": "CreateNewDoc()" },
      { "value": "Open", "onclick": "OpenDoc()" },
      { "value": "Close", "onclick": "CloseDoc()" }
   ]
  }
 }
}
```

2.3 YAML: YAML Ain't Markup Language

```
menu:
 id: file
 value: File
 popup:
  menuitem:
   value: New
   onclick: CreateNewDoc()
  menuitem:
   value: Open
   onclick: OpenDoc()
  menuitem:
   value: Close
   onclick: CloseDoc()
```

3 JSON

3.1 Définition

Un document JSON a pour fonction de représenter de l'information accompagnée d'étiquettes permettant d'en interpréter les divers éléments, sans aucune restriction sur le nombre de celles-ci.

source: wikipedia

3.2 Valeurs

En JSON, il est possible de représenter différents types de données:

- Des nombres: 1, 2, 3, etc...
- du texte (entre double quotes): “element”
- des booléens: true / false

Il est également possible de 'ranger' les éléments sous de formes de tableaux ou de listes.

3.3 Listes

En JSON, les listes seront représentées entre crochets et tous les éléments seront séparés par des virgules.

[“element1”, “element2”, “element3”]

3.4 Tableaux

En JSON, les tableaux sont représentés entre accolades, sous forme de couples clé valeur séparés par deux points. Tous les éléments seront séparés par des virgules.

{”nom”:”toto”, ”prenom”:”titi”, “age”:22, 18:”azerty”}

3.5 Exemples:

En réalité un json sera un ensemble de texte, nombre, listes et tableaux comme le montre les exemples ci-dessous:

ASTUCE: afin de rendre des données au format JSON plus lisible, certains sites tels que http://jsonformatter.curiousconcept.com/ les remette en forme.

ATTENTION: lors du copier coller depuis ce document pdf, les doubles quotes ressortent mal, il faut donc penser à les remplacer par de vrais doubles quotes afin que le site fonctionne correctement.

- Représentation d'une personne:
  - {”nom”:”toto”,”age”:25}

- Représentation d'une liste de personnes:
  - [{"nom":"toto","age":25}, {"nom":"titi","age":50}]
- Il est égalemement possible de représenter une liste en tant que valeur dans un tableau:
  - { "nom":"toto", "jours de travail":["lundi", "mercredi", "vendredi"] }
- etc ...

3.6 A vous de jouer

Afin d'aider la scolaritée à transmettre les notes des étudiants, vous devez écrire ces données au format JSON.

A l'aide du site http://jsonformatter.curiousconcept.com/ créez et validez une série de JSON (Chaque étape inclue les précédentes):

- Un tableaux de matière ayant pour clé le nom de la matière et pour valeur la note (On se contentera des matières suivantes: chimie, biochimie, biologie computationelle).
- Un étudiant avec son nom prénom et les notes précédentes.
- Une classe avec un nom et une liste d'étudiants (Deux ou trois suffiront).

4 Base de données

En général, les données transmises au format JSON ne sont pas écrites à la main mais sont extraites depuis des bases de données. Une base de données est un ensemble de tableaux reliés entre eux.

Nous allons ici utiliser une base de données MySql dans laquelle sont stockées les données concernant les tournages de long métrages de 2002 à 2010 à Paris. Ces données sont issues de la base de données publique suivante `https://www.data.gouv.fr/fr/datasets/lieux-de-tournage-de-films-long-metrage-prs/`.

4.1 Le langage MySQl

Le language utilisé pour requèter une base de données est le language SQL. Ce language est légèrement décliné en fonction de la base de données utilisée.

Dans les scripts et commandes ci-dessous, l'usage des majuscule est une convention plutôt qu'une obligation. Les mots clés (SELECT, FROM, UPDATE, ...) sont mis en majuscules afin de les distinguer parmis les noms de tables et colonnes.

Les trois commandes suivantes seront donc identiques:

- SELECT $*$ FROM etudiants;

- select * from etudiants;
- SeLeCt * FrOm etudiants;

4.2 Installation de la base de données et du seveur web

Lors de l'installation, un écran bleu vous demandera d'entrer un mot de passe. Retenez le bien, il s'agira de votre mot de passe MySql nécessaire juste en dessous.

```
sudo apt-get install apache2 php5 mysql-server libapache2-mod-php5 php5-mysql
```

4.3 Création de la base de données

La commande suivante, permet de créer la base de données et de la remplir avec les informations concernant les tournages de long métrage ayant eu lieu à Paris entre 2002 et 2010.

```
mysql -u root -p < createDatabase.sql
```

Le mot de passe demandé est le mot de passe que vous avez entré lors de l'installation de MySql. A partir de maintenant, ce mot de passe ne vous sera plus demandé. Un utilisateur a également été créé afin de pouvoir débuguer vos scripts sans que vous ayez à nous donnez votre mot de passe:

- Identifiant: user4TpApi
- mot de passe: pass4TpApi

4.4 Utilisation de la base de données

Connectez vous à MySql:

```
mysql -u user4TpApi -p
```

Le mot de passe demandé est pass4TpApi.

4.4.1 Sélection de la base de données

A chaque nouvelle connexion vous devez lancer cette commande.

```
USE databaseTpJson;
```

4.4.2 Requète simple

Cette requète vous permet de récupérer toutes les données depuis la table films.

```
select * from films;
```

4.4.3 Filtre sur les colonnes

Cette requète vous permet de récupérer l'ensemble des titres depuis la table films.

```
select titre from films;
```

4.4.4 Filtre sur les lignes

Cette requète vous permet de récupérer l'ensemble des événements ayant eu lieu dans le 5ème arrondissement.

```
select * from events where arrondissement=75005;
```

N'hésitez pas à jouer et à combiner les différents critères de séléction présentés ci-dessus. Voici la liste des colonnes des différentes tables:

- films:
  - titre
  - realisateur
- events:
  - titre
  - date_debut_evenement
  - date_fin_evenement
  - cadre
  - lieu
  - adresse
  - arrondissement
  - adresse_complete
  - geo_coordinates

4.5 Visualisation des données

Afin de visualiser les données, nous allons utiliser les pages (scripts php) présentes dans le dossier API.

Copiez le dossier API dans /var/www/html avec la commande suivante:

```
sudo cp -R API /var/www/html/.
```

Ces pages, qui renvoient du json, s'appellent des web services. Ouvrez un navigateur web et rendez vous à l'url suivante: `http://localhost/API/films.php` ou `http://localhost/API/events.php`.

Il est également possible de filtrer les événements par film à l'aide de la requete suivante

`http://localhost/events.php?filmId=1`

4.6 Insertion de données

Retournez dans la base de données et insérer un film de test:

```
INSERT INTO films (titre, realisateur) VALUES ('mon super titre', 'mon super
```

Lors de l'insertion, nous ne spécifions pas la valeur de l'id. En effet il s'incrémente automatiquement afin d'avoir un id unique pour chaque donnée. Vous pouvez le vérifier en affichant cette nouvelle entrée depuis la base de données ET en se servant de la page web précédente.

De la même façon, insérez également un nouvel évènement avec au moins une adresse et le filmId correspond au film que vous venez d'insérer. Toutes les autres colonnes sont facultatives. Apès chaque opération, vérifiez que vos données sont correctement insérées.

5 Automatisation

Une fois notre base de données créée, nous souhaitons pouvoir l'interroger depuis un script python.

5.1 Connexion

Pour cet exercice, demandez l'adresse IP de l'un de vos voisins. Pour trouver votre adresse ip, ouvrez un terminal et entrez la commande suivante:

```
ifconfig
```

Cherchez la valeur de inet adr dans le paragraphe wlan0

Dans votre navigateur, interrogez les mêmes pages que précédemment en remplacant localhost par l'adresse ip de votre voisin. Par exemple `http://192.168.1.1/API/films.php`

En faisant cela vous interrogez l'API de votre voisin, ce qui vous permet de récupérer les données présentes dans sa base de données.

5.2 Script

Votre script doit, dans l'ordre, réaliser les étapes suivantes:

- Demander à l'utilisateur une partie de film à chercher (fonction: input("texte d'aide"))
- Lecture de la liste des films correspondants
- Affichage des films et de leur id
- Choix du film par l'utilisateur (fonction: input("texte d'aide"))
- Récupération et affichage des évènements correspondants

AIDE: Vous devez utiliser deux librairies python comme dans ce script d'exemple

```
import json
import urllib2

url = "https://api.ipify.org?format=json"

#urllib2.urlopen permet de récupérer le contenu d'une page web
contenu = urllib2.urlopen(url)

#json.load permet de récupérer le contenu de la page cible sous forme de table
jsonTable = json.load(contenu)

ip = jsonTable['ip']

print ip
```

APPENDICE A: Liste d'exemples d'APIS

- liste d'api scientifiques: `http://www.programmableweb.com/news/195-science-apis-springer-e` `2012/03/28`
- facebook: `https://developers.facebook.com/docs/apis-and-sdks`
- google: `https://developers.google.com/apis-explorer/#p/`
- google maps: `https://developers.google.com/maps/?hl=fr`
- openstreetmap (équivalent Français de google maps): `http://openstreetmap.fr/outils`
- NASA: `https://api.nasa.gov/`
- France: `http://www.opendatafrance.net/`
- Paris:`https://api.paris.fr/`
- SNCF:`https://data.sncf.com/api`
- transport parisien: `http://www.navitia.io/`
- gouvernement américain: `https://www.data.gov/developers/apis`
- `https://www.google.fr/search?q=opendata`