W2 - Colles

W-COL-502

Parsing

Introduction aux REGEX

2.1

EPITECH.

Parsing

binary name: parsing.php
repository name: parsing
language: PHP, CLI, JSON
compilation: none

- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.

Introduction

> "Mince, je ne me souviens plus du nom de ce film, tu sais celui avec le gars là, puis il fait des trucs après..."

Vous avez surement déjà entendu quelqu'un commencer à expliquer un film comme cela. Pour l'aider, vous lui conseillez d'aller sur le site The Movie DB !
Site très pratique qui permet de récupérer plusieurs informations sur des films ou séries. Cependant, les pages de descriptions vous ont toujours parues très denses et les informations importantes ne sont pas assez mises en avant à votre gout. Qu'à cela ne tienne, vous décidez de parser ce site pour récupérer les informations de vos séries et films préférés.
Pour cela, vous allez devoir écrire un script en PHP en utilisant des expressions régulières (REGEX)

Ce projet est ramassé et corrigé par moulinette. Prêtez donc une attention particulière à la rigueur de votre code !

Vous ne pouvez pas utiliser autre chose que des expressions régulières pour ce projet. Toute librairie de parsing de XML ou HTML est rigoureusement interdite.

Vous devez parser les fichiers fournis. Le site TheMovieDB évolue régulièrement, les fichiers fournis correspondent à ceux qui seront testés.

ETAPE 1

La première étape est donc de récupérer le nom et la date de sortie d'un film et créer un fichier JSON avec ces informations. Le fichier JSON doit s'appeller `result.json`.

INFORMATIONS à RéCUPéRER :

- Titre
- Date de sortie

RéSULTAT

```
Terminal
~/W-COL-502> ls
fiche_h2g2.html
~/W-COL-502> php parsing.php fiche_h2g2.html
~/W-COL-502> ls
fiche_h2g2.html
result.json
~/W-COL-502> cat result.json
```

```
{
  "status": "ok",
  "result": {
    "movie": [
      {
        "title": "The Hitchhiker's Guide to the Galaxy",
        "releaseDate": "2005"
      }
    ]
  }
}
```

L'ordre des informations dans le JSON n'a pas d'importance

ETAPE 2

Une fois cette étape passée, vous pouvez retrouver plus d'informations, voici une liste des informations possible de récupérer :

INFORMATIONS à RéCUPERER :

- Résumé
- Statut
- Durée
- Budget
- Revenus
- Langue originale

RéSULTAT

```
{
  "status": "ok",
  "result": {
    "movie": [
      {
        "title": "The Hitchhiker's Guide to the Galaxy",
        "releaseDate": "2005",
        "summary": "Mere seconds before the Earth is to be demolished by an alien
            construction crew, Arthur Dent is swept off the planet by his friend Ford
             Prefect, a researcher penning a new edition of \"The Hitchhiker's Guide
            to the Galaxy.\"",
        "status": "Released",
        "duration": "1h 49m",
        "budget": "$50,000,000.00",
        "revenue": "$104,478,416.00",
        "originalLanguage": "English"
      }
    ]
  }
}
```

L'ordre des informations dans le JSON n'a pas d'importance

ETAPE 3

Vous allez maintenant devoir récupérer les dernières informations. Il s'agit des différents mots clés permettant de définir le film. Vous avez également les genres du film.

INFORMATIONS à RéCUPERER :

- Genres
- Mots clés (Au moins les 5 premiers)
- Acteurs

RéSULTAT

Le fichier JSON de sortie doit avoir ce format :

```
{
  "status": "ok",
  "result": {
    "movie": [
      {
        "title": "The Hitchhiker's Guide to the Galaxy",
        ...,
        "genre": [
          "Adventure",
          "Comedy",
          "Family",
          "Science Fiction"
        ],
        "keywords": [
          "bureaucracy",
          "android",
          "based on novel or book",
          "chase",
          "space travel"
        ],
        "cast": [
          { "name": "Martin Freeman" , "character": "Arthur Dent" },
          { "name": "Zooey Deschanel" , "character": "Trillian" },
          { "name": "Sam Rockwell" , "character": "Zaphod Beeblebrox" },
          { "name": "Yasiin Bey" , "character": "Ford Prefect" },
          { "name": "John Malkovich" , "character": "Humma Kavula" }
        ]
      }
    ]
  }
}
```

L'ordre des informations dans le JSON n'a pas d'importance

ETAPE 4

Vous allez maintenant vous rendre compte que une série et un film c'est la même chose sur ce site. Vous allez donc devoir parser aussi les séries. Cependant, vous allez devoir les mettre dans un autre objet que l'objet `movie`, un nouvel objet `tv` :

INFORMATIONS à RéCUPERER :

- Les même que pour un film mais dans un objet nommé `tv`

RéSULTAT

Le fichier JSON de sortie doit avoir ce format :

Terminal

```
~/W-COL-502> ls
fiche_batman.html
~/W-COL-502> php parsing.php fiche_batman.html
~/W-COL-502> ls
fiche_batman.html
result.json
~/W-COL-502> cat result.json
```

```
{
  "status" : "ok",
  "result": {
    "tv" : [
      {
        "title": "Batman: The Animated Series",
        ...
        "cast": [
          { "name": "Kevin Conroy", "character": "Batman/Bruce Wayne" },
          { "name": "Efrem Zimbalist Jr.", "character": "Alfred Pennyworth" },
          { "name": "Bob Hastings", "character": "Commissioner James Gordon" },
          { "name": "Loren Lester", "character": "Robin/Dick Grayson" },
          { "name": "Robert Costanzo", "character": "Det. Harvey Bullock" }
        ]
      }
    ]
  }
}
```

L'ordre des informations dans le JSON n'a pas d'importance

ETAPE 5

Maintenant que votre programme prend en compte les séries et les films, on doit pouvoir lui donner plusieurs fiches et il doit être capable de retrouver toutes les informations. En regroupant les films entre eux et les séries entre elles.

RÉSULTAT

Le fichier JSON de sortie doit avoir ce format :

Terminal

```
~/W-COL-502> ls
fiche_batman.html
fiche_h2g2.html
~/W-COL-502> php parsing.php fiche_batman.html fiche_h2g2.html
~/W-COL-502> ls
fiche_batman.html
result.json
~/W-COL-502> cat result.json
```


```
{
  "status": "ok",
  "result": {
    "movie": [
      {
        "title": "The Hitchhiker's Guide to the Galaxy",
        "releaseDate": "2005",
        "summary": "Mere seconds before the Earth is to be demolished by an alien
            construction crew, Arthur Dent is swept off the planet by his friend Ford
             Prefect, a researcher penning a new edition of \"The Hitchhiker's Guide
            to the Galaxy.\"",
        "status": "Released",
        ...,
      }
    ],
    "tv": [
      {
        "title": "Batman: The Animated Series",
        "releaseDate": "1992",
        "summary": "Batman: The Animated Series is an American animated television
            series based on the DC Comics superhero Batman. The series was widely
            praised for its thematic complexity, dark tone, artistic quality, and
            faithfulness to its title character's crime-fighting origins.",
        "status": "Ended",
        "duration": "30m, 22m",
        "originalLanguage": "English",
        "genre": [
          "Action &amp;amp; Adventure",
          "Animation",
          "Drama",
          "Mystery"
        ],
        "keywords": [
```

```
            "dc comics",
            "superhero",
            "based on comic",
            "robin",
            "super power"
          ],
          "cast": [
            { "name": "Kevin Conroy", "character": "Batman/Bruce Wayne" },
            { "name": "Efrem Zimbalist Jr.", "character": "Alfred Pennyworth" },
            { "name": "Bob Hastings", "character": "Commissioner James Gordon" },
            { "name": "Loren Lester", "character": "Robin/Dick Grayson" },
            { "name": "Robert Costanzo", "character": "Det. Harvey Bullock" }
          ]
        }
      ]
    }
}
```

L'ordre des informations dans le JSON n'a pas d'importance

Dans le cas où la donnée n'est pas présente dans le HTML, il ne faut pas la mettre dans le JSON