

DATA SCRAPING

Initiation

DATA SCRAPING

Le web scraping est une technique d'extraction du contenu de sites Web

DATA SCRAPING

Pourquoi faire ?

Exemple d'utilisation:

Bot / web crawler (Robots des Moteurs de recherche)

Comparateur de prix

Récupération de contact

Observation Météo

Modification de site web

Web Meshup

...

Récupérer des données sur Internet

Les techniques



Le bon vieux

COPIER

COLLER

Photo par Andreas Müller

Liste des techniques:

L'utilisation de **GREP** sur du texte et les expressions régulières

Programmation **HTTP**

Les **HTML Parser** (analyse syntaxique du HTML)

Analyse du **DOM** (Document Object Model)

Des logiciels de scraping

Des plateformes d'agrégation verticale

L'utilisation des **metadatas** et des **microformats**

Analyse par vision informatique d'une page (Machine learning)

Source wikipedia

Les outils

Les outils

Apache Camel - Archive.is - Automation Anywhere -
Convertigo - **cURL** - Data Toolbar - Diffbot - **Firebug 2.23** -
Greasemonkey - Heritrix - HtmlUnit - HTTrack - iMacros -
Import.io - Jaxer - **Node.js** - nokogiri - OutWit Hub -
PhantomJS - ScraperWiki - Scrapy - Selenium - SimpleTest -
UiPath - watir - **Wget** - **Wireshark** - WSO2 Mashup Server -
Yahoo! Query Language (YQL)

Source wikipedia

Tout langage de programmation

L'un des plus utilisés



Comprenons la finalité
pour mieux l'utiliser

Le bon scraper:
Il charge des données et les scrap

Le mauvais scraper:
Il charge des données et les scrap

Mais ...

DATA SCRAPPING

=

ETL

Extract **T**ransform **L**oad

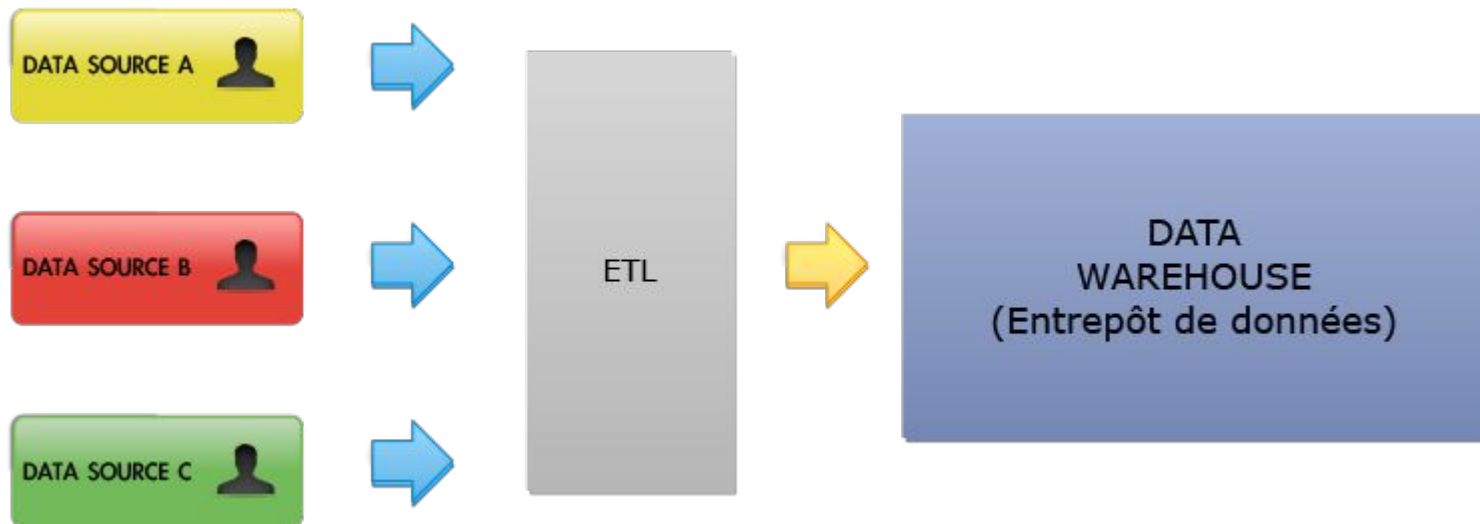
ETL

La nourricière du Big Data

EXTRACT

TRANSFORM

LOAD





FINANCE

MARKETING

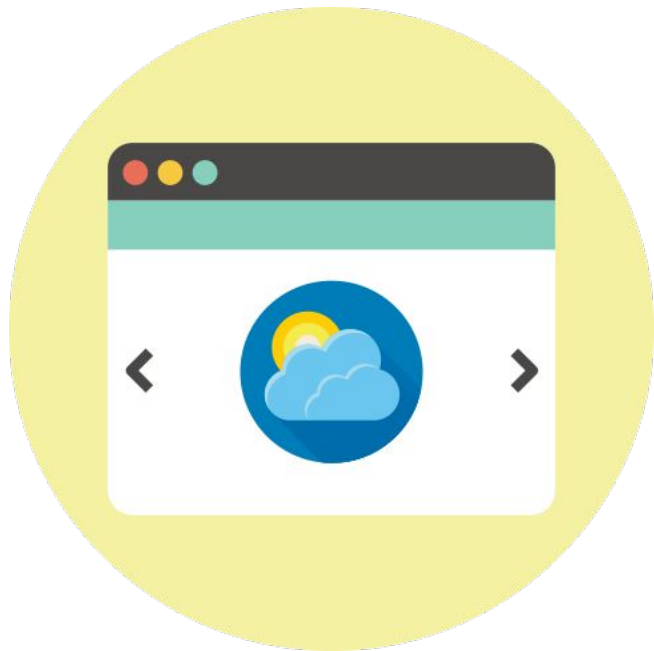
VENTE

Hadoop / Cube / Hana



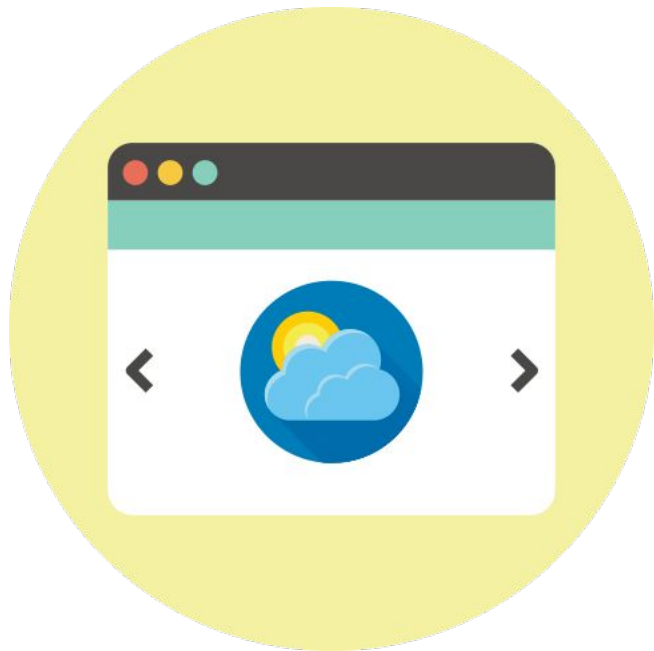
Reporting Data Viz

Ca fait beaucoup de théorie.
Revenons à nos moutons



Exemple

Je veux extraire la météo d'un site pour l'afficher sur mon site web.



Les données désirées:

La température, le pictogramme

Un petit effort et j'inclus:

Les méta données:

Heure de l'observation

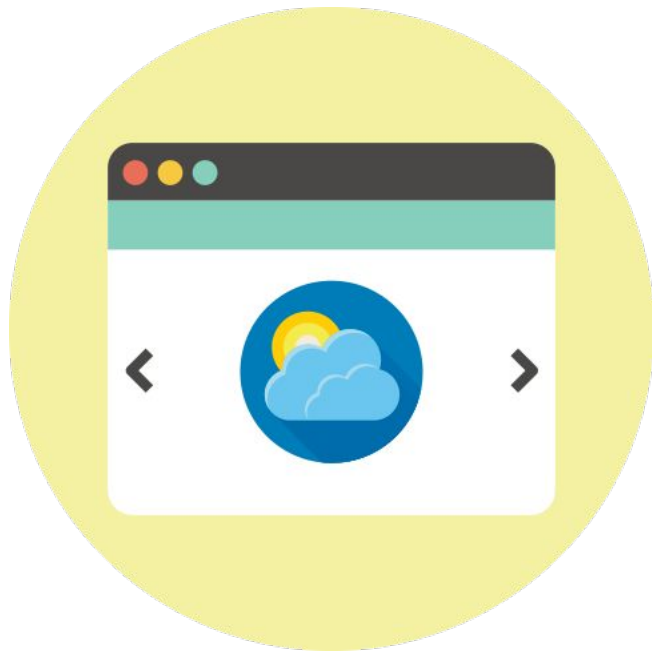
Heure de la saisie

Et des données connexes comme:

Le commentaire textuel

Les prévisions

...



Et comme je suis un pro

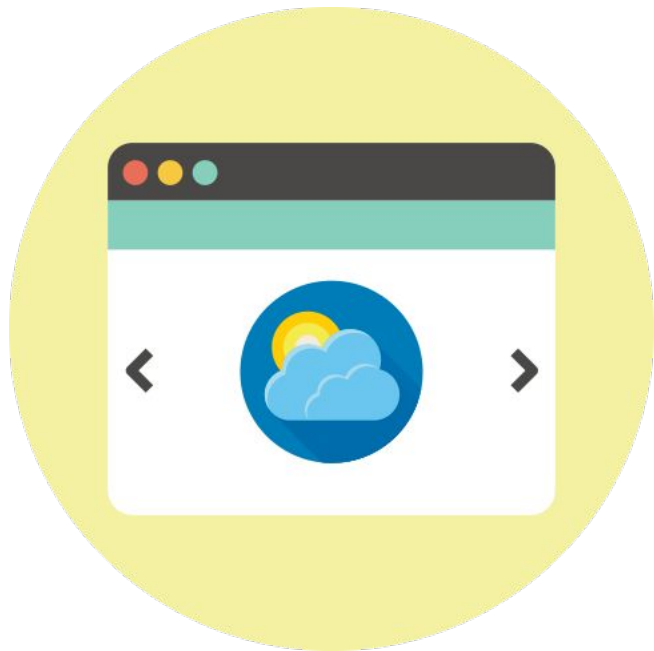
J'ai un document de définition qui indique:

L'anatomie des URL de mes extractions

La fréquence d'extraction

Une description du format de donnée
(en entrée et sortie)

...



Bon c'est bien joli tout ça
Mais concrètement on fait comment ?

Commençons simplement

Récupérons les données du site Vigicrues

Un site du réseau developpement-durable.gouv.fr





Information nationale ▾ Informations locales ▾

■ Accueil > Informations locales > Rhône amont-Saône > Données temps réel : Besançon

Graphique Tableau Infos station

Besançon (Doubs) - Hauteurs en m

Date	Besançon
20/06/2016 16:00	3.59
20/06/2016 15:00	3.6
20/06/2016 14:00	3.61
20/06/2016 13:00	3.62
20/06/2016 12:00	3.63
20/06/2016 11:00	3.65
20/06/2016 10:00	3.68

Ouvrez l'URL suivant:

<http://import.io>

Click to go back, hold to see history

[Product](#) [Pricing](#) [Partners](#) [Blog](#) [Help](#) [Contact sales](#)

[Log in](#) or [Sign up](#)

Extract web data the easy way

Drive data insight with the world's #1 web data platform.

 Enter a URL for a page with data

[Try it out](#)




[Request free trial](#)

[See some examples](#)



import.io [Give feedback](#) [Help](#)

Un site du réseau developpement-durable.gouv.fr

Information nationale Informations locales

Accueil > Informations locales > Rhône amont-Saône > Données temps réel : Besançon

Graphique Tableau Infos station

Besançon (Doubs) - Hauteurs en m

Date	Besançon
20/06/2016 22:00	3.51
20/06/2016 21:00	3.52
20/06/2016 20:00	3.54
20/06/2016 19:00	3.56
20/06/2016 18:00	3.57
20/06/2016 17:00	3.57
20/06/2016 16:00	3.59
20/06/2016 15:00	3.6
20/06/2016 14:00	3.61
20/06/2016 13:00	3.62
20/06/2016 12:00	3.63
20/06/2016 11:00	3.65
20/06/2016 10:00	3.68
20/06/2016 09:00	3.7
20/06/2016 08:00	3.71
20/06/2016 07:00	3.72
20/06/2016 06:00	3.74
20/06/2016 05:00	3.75
20/06/2016 04:00	3.75
20/06/2016 03:00	3.77
20/06/2016 02:00	3.77
20/06/2016 01:00	3.77
20/06/2016 00:00	3.78
19/06/2016 23:00	3.79

We're working on grabbing data from your page...

...it will only take a few seconds.

import.io

[Give feedback](#) [Help](#)

Add or manage URLs

Create a blank table

Undo

Redo

Done

Data view

#	Date value	Besanon number
1	20/06/2016 22:00	3.51
2	20/06/2016 21:00	3.52
3	20/06/2016 20:00	3.54
4	20/06/2016 19:00	3.56
5	20/06/2016 18:00	3.57
6	20/06/2016 17:00	3.57
7	20/06/2016 16:00	3.59
8	20/06/2016 15:00	3.6

[Pricing](#)[Help](#)[Dashboard](#)[My account](#) ▾

www.vigicrues.gouv.fr

[↻ Run URLs](#)[📄 Download CSV](#)[✎ Edit](#)[📄 Duplicate](#)[🗑 Delete](#)[Settings](#)[Run history](#)[Integrate](#)

Live query API

Use the following RESTful API to query your Extractor live. Note that you can only query one URL at a time with this API. Please do not share your API key with anyone.

```
https://extraction.import.io/query/extractor/8c9fb889-0318-422a-88a9-bcc68dd10426?_apikey=62515a04568a4defaa03cc073a105bff024730fffa294a24f37481fabee6037cbc390ec9e58dceeb050b5e3fe8f8592d7b9facde5901b8a5aa0a9722b1270abe319a756d407e4e180d5852e8bc6dbc24&url=http%3A%2F%2Fwww.vigicrues.gouv.fr%2Fniveau3.php%3FCdStationHydro%3DU251201001%26typegraphe%3Dh%26AffProfondeur%3D168%26nbrstations%3D9%26ong%3D2%26Submit%3DRefaire%2Ble%2Btableau%2B-%2BValider%2Bla%2Bs%25C3%25A9lection
```



```
1 {
2   "extractorData" : {
3     "url" : "http://www.vigicrues.gouv.fr/niveau3.php?CdStationHydro=U25:",
4     "resourceId" : "9a45cac767ec4c711fa3232ca53cad63",
5     "data" : [ {
6       "group" : [ {
7         "Date value" : [ {
8           "text" : "20/06/2016 21:00"
9         } ],
10        "Besanon number" : [ {
11          "text" : "3.52"
12        } ]
13      }, {
14        "Date value" : [ {
15          "text" : "20/06/2016 20:00"
16        } ],
17        "Besanon number" : [ {
18          "text" : "3.54"
19        } ]
20      }, {
21        "Date value" : [ {
22          "text" : "20/06/2016 19:00"
23        } ],
```

Oui mais !!!

**C'est pas Open Source
C'est payant
Et je suis un pro !**



Voyons comment écrire cela en PHP



Voici la recette

Un Extracteur d'url:

simplifiez vous la vie, utilisez [file_get_contents\(\)](#)

Un Parser de page web:

Profitez de la puissance de [DOM en PHP](#)

On met tout ça dans une boucle, on arrose de ; et le tour est joué.
Vous obtiendrez votre premier “Data scrapping” en 20 lignes* de PHP.

* Sans les commentaires

Le plus simple l'extracteur

```
1 <?php
2
3 // setting the URL
4 $file = "http://www.vigicrues.gouv.fr/mes_parametre_de_requete?...";
5
6 // getting the URL
7 $content = file_get_contents($file);
8
```

Notez que pour des requêtes plus complexes nous pouvons utiliser cURL

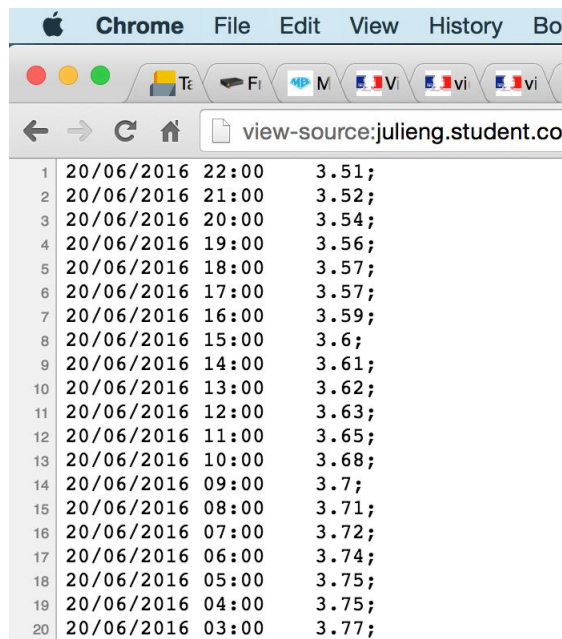
Un peu plus complexe DOM

```
10
11 // Create a dom Document object
12 $doc = new DOMDocument();
13
14 // Load content in the DOM
15 $doc->loadHTML($content);
16
17 // Finding first table
18 $table = $doc->getElementsByTagName('table')->item(0);
19 // Extract all rows in the table
20 $rows = $table->getElementsByTagName("tr");
21
```

Enfin la boucle qui récupère les données

```
22
23 $data = "";
24
25 foreach ($rows as $row)
26 {
27
28     // for each row (TR) get all cells (TD)
29     $cells = $row->getElementsByTagName('td');
30
31     // First line include table header and there is no TD on it
32     // So if there is no cells, we skip the line using continue
33     if($cells->length == 0) { continue; }
34
35     // For each cell get the values
36     foreach ($cells as $cell)
37     {
38         // adding the value to data and separate the value with a TAB (\t)
39         $data .= $cell->nodeValue."\t";
40     }
41
42     // remove the last unneeded TAB (\t)
43     $data = rtrim($data,"\t");
44
45     // Adding a semicolon and a carriage return at the end of line
46     $data .= ";\n";
47
48 }
49
50
51 echo $data;
52
```

Nous obtenons un CSV bien propre



The screenshot shows a Chrome browser window with the address bar displaying 'view-source:julieng.student.co'. The page content is a CSV file with 20 rows of data. Each row contains a date and time in 'DD/MM/YYYY HH:MM' format, followed by a numerical value. The data is presented in a table-like structure with line numbers on the left.

1	20/06/2016 22:00	3.51;
2	20/06/2016 21:00	3.52;
3	20/06/2016 20:00	3.54;
4	20/06/2016 19:00	3.56;
5	20/06/2016 18:00	3.57;
6	20/06/2016 17:00	3.57;
7	20/06/2016 16:00	3.59;
8	20/06/2016 15:00	3.6;
9	20/06/2016 14:00	3.61;
10	20/06/2016 13:00	3.62;
11	20/06/2016 12:00	3.63;
12	20/06/2016 11:00	3.65;
13	20/06/2016 10:00	3.68;
14	20/06/2016 09:00	3.7;
15	20/06/2016 08:00	3.71;
16	20/06/2016 07:00	3.72;
17	20/06/2016 06:00	3.74;
18	20/06/2016 05:00	3.75;
19	20/06/2016 04:00	3.75;
20	20/06/2016 03:00	3.77;

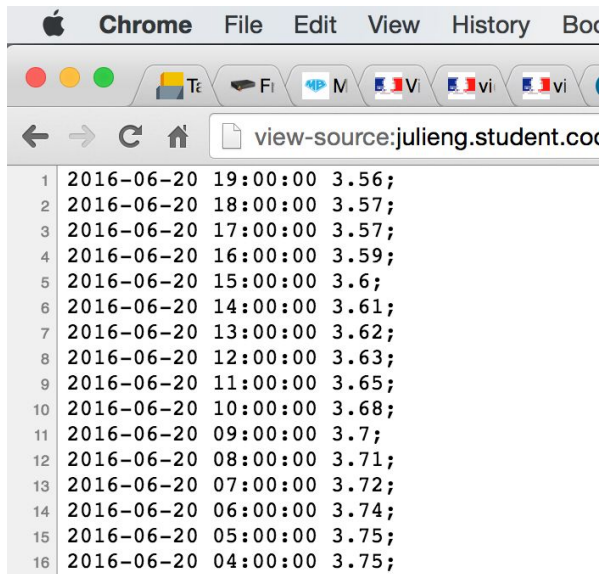
Un peu de transform

La date n'est pas dans un format très exploitable

Un peu de transform

```
33 // For each cell get the values
34 foreach ($cells as $cell)
35 {
36     // Getting cell Value
37     $cellValue = $cell->nodeValue;
38
39     // If the cell value size match the size of a date
40     if(strlen($cell->nodeValue) == 16)
41     {
42         // We create a MySQL date format with the date
43         // ie. 2016-06-20 18:00 / YYYY-MM-DD HH:MM
44
45         // Thanks to PHP we can get a date from a string
46         $dateObject = date_create_from_format('d/m/Y H:i',$cellValue);
47
48         // We use the date with our wanted format
49         $cellValue = date_format($dateObject,'Y-m-d H:i:s');
50
51     }
52
53     // adding the value to data and separate the value with a TAB (\t)
54     $data .= $cellValue."\t";
55 }
```

Notre date est maintenant au format MySQL



1	2016-06-20 19:00:00 3.56;
2	2016-06-20 18:00:00 3.57;
3	2016-06-20 17:00:00 3.57;
4	2016-06-20 16:00:00 3.59;
5	2016-06-20 15:00:00 3.6;
6	2016-06-20 14:00:00 3.61;
7	2016-06-20 13:00:00 3.62;
8	2016-06-20 12:00:00 3.63;
9	2016-06-20 11:00:00 3.65;
10	2016-06-20 10:00:00 3.68;
11	2016-06-20 09:00:00 3.7;
12	2016-06-20 08:00:00 3.71;
13	2016-06-20 07:00:00 3.72;
14	2016-06-20 06:00:00 3.74;
15	2016-06-20 05:00:00 3.75;
16	2016-06-20 04:00:00 3.75;

Finissons avec un Load

Sur le même script

```
3 // Specify the Station in the query
4 $station = "U251201001";
```

On ajoute l'identifiant de la station

```
25
26 $SQL = "";
27
28 foreach ($rows as $row)
29 {
```

On renomme \$data \$SQL pour plus de cohérence

Finissons avec un Load

On crée pour chaque “rows” une requete d’insertion SQL

```
36 // Setting a SQL statement String
37
38 // Setting a SQL statement String
39 $SQL .= "INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES(";
40
41 // For each cell get the values
42 foreach ($cells as $cell)
43 {
```

Enfin on formate simplement les valeurs pour la requête SQL

```
60
61 // adding the value to the SQL statement , protecting with ' and adding a comma
62 $SQL .= "".$cellValue."',";
63 }
64
65 // Finishing the SQL Statement and adding a carriage return at the end of line
66 $SQL .= "".$station."');\n";
```

On obtient une chaine SQL prête à être importée

```
1 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 19:00:00','3.56','U251201001');
2 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 18:00:00','3.57','U251201001');
3 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 17:00:00','3.57','U251201001');
4 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 16:00:00','3.59','U251201001');
5 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 15:00:00','3.6','U251201001');
6 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 14:00:00','3.61','U251201001');
7 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 13:00:00','3.62','U251201001');
8 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 12:00:00','3.63','U251201001');
9 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 11:00:00','3.65','U251201001');
10 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 10:00:00','3.68','U251201001');
11 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 09:00:00','3.7','U251201001');
12 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 08:00:00','3.71','U251201001');
13 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 07:00:00','3.72','U251201001');
14 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 06:00:00','3.74','U251201001');
15 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 05:00:00','3.75','U251201001');
16 INSERT INTO vigicrue_data (observation_date,observation_value,station_id) VALUES('2016-06-20 04:00:00','3.75','U251201001');
```

Dans la vie réelle on ne procède pas ainsi

**Les trois étapes devraient être séparées.
Mais le concept est là:**

**J'extrait (Extract)
Je transforme (Transform)
Je charge (Load)**

Notre exemple soulève un problème

La temporalité



Les données vivent

Elles ont un cycle, il faut suivre ce cycle:

Récupérer les nouvelles actualisation

Mais aussi, archiver les informations désuètes.

Relancer un script à interval régulier

En informatique nous utilisons des “Tâches planifiées”.

Sous Linux et Unix un utilitaire gère ce besoin

CRONTAB

Nous appelons les “appels” des tâches cron

Voici un résumé

Pour éditer une tâche cron nous utilisons la commande:

crontab -e

Une tâche est configurée en ajoutant ce type de ligne:

01 04 * * * /usr/bin/somedirectory/somecommand

Cette ligne indique d'exécuter "somecommand" tous les jours à 4H01 du matin

Pour plus de détail consultez:

wikipedia.org/wiki/Cron

Un exemple de barbare

C'est brut mais efficace

Nous récupérons l'url avec CURL et nous filtrons avec GREP

Par exemple je veux voir les metas d'une page:

```
curl http://www.ikea.com/fr/fr/catalog/products/S29097737/ | grep meta
```

Grep supporte les expressions régulières, filtrons tous les urls d'une page

```
curl http://www.ikea.com/fr/fr/catalog/products/S29097737/ | grep -o 'http://[a-zA-Z0-9.-]*/'
```

Le tout peut être sauvé dans un fichier en ajoutant

>monfichier.txt

à la fin de la commande

Une approche différente



PhantomJS

PhantomJS

Un outils de scraping Javascript

**Il permet d'interagir en JS avec les pages
et bien plus encore**



PhantomJS

PhantomJS

Récupérer le titre d'une page

```
1 var webPage = require('webpage');
2 var page = webPage.create();
3
4 ▼ page.open('http://www.ikea.com/fr/fr/catalog/products/S29097737/', function(status) {
5
6 ▼   var title = page.evaluate(function() {
7     return document.meta;
8   });
9
10   console.log(title);
11   phantom.exit();
12
13 ~ });
```



PhantomJS

PhantomJS

Récupérer les infos avec JQuery

```
1 "use strict";
2 var page = require('webpage').create();
3
4 page.onConsoleMessage = function(msg) {
5     console.log(msg);
6 };
7
8 page.open("http://www.accesscodeschool.fr/blog/", function(status) {
9     if (status === "success") {
10         page.includeJs("http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js", function() {
11             page.evaluate(function() {
12                 console.log($("#H2").first().text());
13             });
14             phantom.exit(0);
15         });
16     } else {
17         phantom.exit(1);
18     }
19 });
```



PhantomJS

PhantomJS

Créer une image ou un PDF d'une page web

Utilisez le script convert_page.js ainsi:

```
phantomjs convert_page.js http://www.google.com google.png
```

ou

```
phantomjs convert_page.js http://www.google.com google.pdf
```



PhantomJS

PhantomJS

**Un super outil très utilisé.
Plein d'exemple et de ressource
RTFM !!!**



PhantomJS

Fichiers d'exemples disponible sur:

<https://github.com/jubry/ACS>



Play Time !!!

