

Mobilité, multimédia, données ouvertes

Guillaume Allègre
Guillaume.Allegre@silecs.info

Polytech Grenoble - RICM3

2018

Données ouvertes

Présentation générale

Données ouvertes

- ▶ Intervenants
 1. **Données ouvertes (DO)** : Guillaume Allègre
 2. Mobilité, multimédia : Sébastien Pittion
- ▶ 8 cours (mercredi 10h15)
 1. 4 cours DO (Web, PHP, données ouvertes)
 2. 4 cours MM (développement mobile)
- ▶ 6 séances de TP (jeudi 8h)
 - ▶ 3 séance DO : web, PHP
 - ▶ 3 séances MM

Données ouvertes ?

- ▶ "Open Data"
 1. ex. <https://www.data.gouv.fr/fr>
Plateforme ouverte des données publiques françaises
 2. ex. <http://data.beta.metropolegrenoble.fr>
Portail Métro - Grenoble - SMTC
- ▶ Données collaboratives
 - ▶ **Wikidata**
 - ▶ OpenStreetMap (BDD cartographique)
 - ▶ données ROMMA (Réseau Observation Météo du Massif Alpin)
 - ▶ The Movie Database (TMDb)
 - ▶ ...
- ▶ Dans tous les cas
 - ▶ une structure d'interrogation (API)

"Données ouvertes" - Open data

► Définition

Données qu'un organisme met à la disposition de tous sous forme de fichiers numériques afin de permettre leur réutilisation.

► Notes

1. Les données ouvertes n'ont généralement pas de caractère personnel.
2. Elles sont accessibles dans un format favorisant leur réutilisation.
3. La réutilisation des données ouvertes peut être soumise à conditions.

► Ouverture des données

Politique par laquelle un organisme met à la disposition de tous des données numériques, dans un objectif de transparence ou afin de permettre leur réutilisation, notamment à des fins économiques.

(J.O. du 3 mai 2014)

Données ouvertes - édition collaborative

- ▶ Contributions
 - ▶ Ouvertes à tous
 - ▶ Sans discrimination (âge, sexe, localisation, culture...)
 - ▶ on parle de **crowdsourcing**

- ▶ Licence ouverte pour les données
 - ▶ Paternité des contributions
 - ▶ Enrichissement d'un bien commun

- ▶ Usage libre
 - ▶ Services dérivés
 - ▶ Services innovants

Données ouvertes : des licences adaptées

- ▶ Similaire aux licences libres du logiciel libre
 - ▶ licences BSD
 - ▶ licences "**copyleft**", ex. GPL (General Public License)
- ▶ Licences "Open data" (souvent non copyleft)
 - ▶ **LO/OL** Licence ouverte / Open licence (gouv.fr 2011)
 - ▶ **OGL** Open Government Licence (UK gov. 2010)
- ▶ Licences collaboratives (souvent copyleft)
 - ▶ **ODbL** Open Database License (OSM, Open Food Facts) (OKF)
 - ▶ **CC0** presque "Domaine public" (Wikidata)
 - ▶ **CC By-SA** Creative Commons - Attribution - Share alike (Wikipedia)

Technologies

- ▶ Technologies web
 - ▶ côté serveur : **PHP**, Java, Ruby, Python, ...
 - ▶ côté client : navigateurs, **Javascript**, ...
- ▶ Protocole HTTP : très flexible
- ▶ Changements d'utilisation
 - ▶ HTML (humain) → XML, JSON (machine)
 - ▶ navigateur → robots (scripts), applications (mobiles)
- ▶ Technologies "à la mode"
 - ▶ *Dataviz* (visualisation), *Big data*
 - ▶ aspects moins théoriques

Programme

- ▶ TP 1 - Introduction aux technologies web
 - ▶ HTML (bases)
 - ▶ Protocoles web
 - ▶ PHP - coté serveur

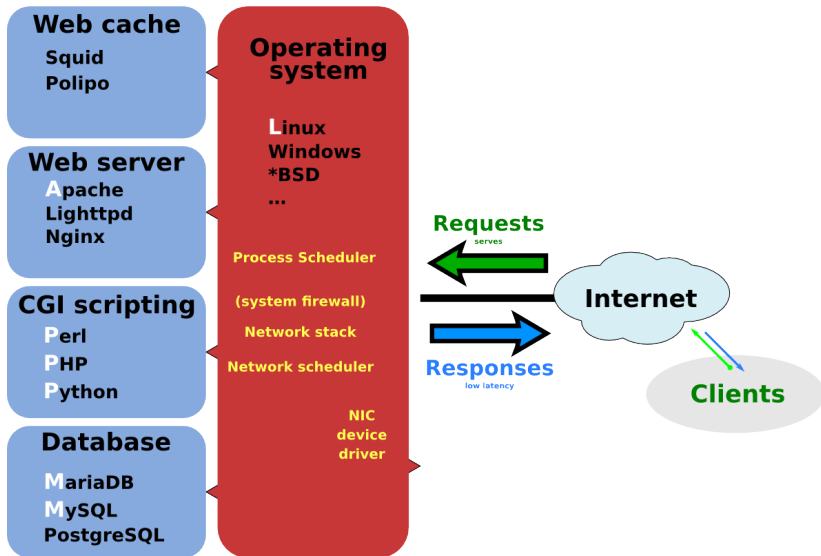
- ▶ TP 2 - Données ouvertes et webservice
 - ▶ les API web
 - ▶ JSON, XML
 - ▶ service TMDb

- ▶ TP 3 - Wikidata et SPARQL
 - ▶ le web sémantique
 - ▶ Wikidata
 - ▶ SPARQL

Architecture web

Architecture web classique - LAMP

by Shmuel Csaba Otto Traian; CC BY-SA 4.0; created 2013-09-10, last updated 2014-03-28



Technologies web

- ▶ Un protocole
 - ▶ **HTTP** HyperText Transfer Protocol
 - ▶ **HTTPS** variante chiffrée / signée

- ▶ Des contenus
 - ▶ **HTML** HyperText Markup Language : format des "pages web"
 - ▶ tout autre contenu accessible par URL (images, binaires, etc.)

Le protocole HTTP

Discussion HTTP

Exemple de requête HTTP

GET /page-test.html HTTP/1.1

Host : www.exemple.fr

User-Agent : Opera/9.22 (X11; Linux x86_64; U; en)

Accept : text/html, application/xml;q=0.9

Accept-Charset : iso-8859-1, utf-8

Discussion HTTP

Exemple de requête HTTP

GET /page-test.html HTTP/1.1

Host : www.exemple.fr

User-Agent : Opera/9.22 (X11; Linux x86_64; U; en)

Accept : text/html, application/xml;q=0.9

Accept-Charset : iso-8859-1, utf-8

Exemple de réponse HTTP

HTTP/1.1 200 OK

Date : Tue, 13 Nov 2007 10 :32 :48 GMT

Server : Apache/2.0.52 (CentOS)

Accept-Ranges : bytes

Connection : close

Content-Type : text/html; charset=UTF-8

... *[HTML de la page]* ...

Caractéristiques du HTTP

Un dialogue client-serveur est formé de :

- ▶ Requête HTTP
 - ▶ source d'informations variées :
IP, version du navigateur, OS, langue, page de provenance, etc.
- ▶ Réponse HTTP
 - ▶ entête + contenu
 - ▶ contenu pas forcément en HTML : Content-Type (MIME)
 - ▶ permet la redirection d'URL

Caractéristiques du HTTP

Un dialogue client-serveur est formé de :

- ▶ Requête HTTP
 - ▶ source d'informations variées :
IP, version du navigateur, OS, langue, page de provenance, etc.
- ▶ Réponse HTTP
 - ▶ entête + contenu
 - ▶ contenu pas forcément en HTML : Content-Type (MIME)
 - ▶ permet la redirection d'URL

Exemple de réponse : Redirection HTTP

HTTP/1.x 301 Moved Permanently

Date : Tue, 13 Nov 2007 10 :32 :48 GMT

Location : <http://newserver.fr/page.php>

Bases de HTTP : les méthodes

- ▶ Principales méthodes HTTP
 - ▶ **GET** récupérer une ressource (la plus fréquente)
 - ▶ **POST** envoi d'un bloc de données (formulaires, uploads)
 - ▶ **HEAD** comme GET, mais limitée à l'en-tête (tests)
 - ▶ **PUT** envoi d'une ressource pour stockage (upload)
 - ▶ **DELETE** suppression d'une ressource du serveur (rare)

- ▶ Notes
 - ▶ Méthodes "sûres" (safe) : GET, HEAD.
Par principe, ne doivent rien modifier sur le serveur.
 - ▶ Méthodes "idempotentes" : GET, HEAD, PUT, DELETE.
L'envoi répété donne le même résultat qu'un envoi unique.

La méthode GET

- ▶ Méthode GET

- ▶ l'URL (Uniform Resource Locator)
- ▶ protocole (http)
- ▶ nom de serveur (canonique ou alias DNS)
- ▶ ressource
- ▶ paramètres

- ▶ Exemples

Dans un navigateur (Chrome/Firefox/Opéra), outils développeur (Ctrl+Shift+I), onglet Réseau.

Anatomie d'une URL

`http://www.exemple.fr:8080/chemin/d/acces.php?q=v1&q2=v2#ancree`

- ▶ `http` protocole, en web `http` ou `https`
- ▶ `www.exemple.fr` serveur web contacté
- ▶ `8080` port IP ; par défaut 80 (`http`) ou 443 (`https`)
- ▶ `chemin/d/acces.php` chemin de la ressource sur le serveur
- ▶ `q=v1, q2=val2` paramètres de **requête** (méthode GET)
- ▶ `ancree` repère de la ressource à l'intérieur de la page

Terminologie

- ▶ **URL** : Uniform Resource Locator
- ▶ URN : Uniform Resource Name ex. `urn:isbn:0-395-36341-1`
- ▶ URI : Uniform Resource Identifier, regroupe les URL et URN

Les principaux codes diagnostic

- ▶ 1xx Informational
 - ▶ 100 Continue
- ▶ 2xx Success
 - ▶ 200 OK
 - ▶ 206 Partial content
- ▶ 3xx Redirection
 - ▶ 301 Moved permanently
 - ▶ 303 See other
 - ▶ 307 Temporary redirect
- ▶ 4xx Client Error
 - ▶ 403 Forbidden
 - ▶ 404 Not found
- ▶ 5xx Server Error
 - ▶ 500 Internal Server error
 - ▶ 503 Service Unavailable

HTML - Hypertext Markup Language

Exemple type (html 5)

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Horloge</title>
  </head>

  <body>
    <p>
      Nous sommes le lundi 12 janvier 2054.<br />
      Il est <strong>10 h 15</strong>.
    </p>
  </body>
</html>
```

Quelques points de HTML - 1/2

- ▶ Structure générale d'arbre
 - ▶ `<html>`
 - ▶ `<head>`
 - ▶ `<body>`

- ▶ Vocabulaire
 - ▶ **éléments** et balises (tags), ex. ``
 - ▶ balises ouvrantes/fermantes ou autofermantes,
 - ▶ `ce document`
 - ▶ `
`
 - ▶ attributs et valeurs
 - ▶ attribut src : ``
 - ▶ valeur libre (src) ou imposée (liste d'autorité)

Quelques points de HTML - 2/2

▶ Quelques éléments

- ▶ Titres `<h1>` ... `<h6>`
- ▶ Lien et ancre ``
- ▶ `<div>` division (section) arbitraire bloc
- ▶ `` passage arbitraire en ligne (*inline*)

▶ Quelques attributs

- ▶ `id="myIdentifiant"` identifier unique d'un élément
- ▶ `class="citation"` classe d'un élément (multiple)
- ▶ `title="des précisions..."` une infobulle (à la souris)

Formulaire HTML

```
<form name="ex" action="suite.php" method="get">
  <fieldset>
    <label>Login :</label>
    <input name="login" type="text" />
    <label>Mot de passe :</label>
    <input name="mdp" type="password" />
    <label>Choix :</label>
    <select name="choix">
      <option selected="selected">1</option>
      <option value="2">deuxieme</option>
    </select>
    <button type="submit">OK</button>
  </fieldset>
</form>
```

HTML : historique

- ▶ Origines
 - ▶ langage texte balisé **simple**
 - ▶ HyperText Markup Language (CERN 1990s)
 - ▶ héritier du SGML (Standard Generalized Markup Language)

- ▶ Évolutions...
 - ▶ progressivement standardisé et "nettoyé" (W3C)
 - ▶ a motivé l'élaboration du XML (eXtensible Markup Language), 1998
 - ▶ (comme XML) implémente une structure d'**arbre**
 - ▶ associé à d'autres normes : CSS, JavaScript

Les nouveautés de HTML5

- ▶ Objets "multimédia"
 - ▶ élément `<audio>`
 - ▶ élément `<video>`
 - ▶ élément `<canvas>` + interactions JS
- ▶ Amélioration des documentations
 - ▶ Documentation API JavaScript/DOM de référence
 - ▶ Documentation aussi orientée développeur client web
 - ▶ HTML :
<http://www.w3.org/TR/html/Overview.html#contents>
- ▶ Effet de bord : obsolescence de la branche XHTML
 - ▶ branche XHTML (HTML sur XML) : standard théorique "pur"
 - ▶ HTML4, 5... : standard implémenté

En résumé : Les contenus web

- ▶ Le web des humains
 - ▶ HTML classique (document)
 - ▶ tous fichiers binaires (images...)
 - ▶ fichiers texte "web" : CSS, Javascript
 - ▶ transfert de fichiers (download, upload)

- ▶ Le web des machines
 - ▶ XML (RSS, Atom, ...)
 - ▶ JSON (JavaScript Object Notation)
 - ▶ entre serveurs et clients (widgets)
 - ▶ entre serveurs (API web)

Les feuilles de style CSS

Introduction aux CSS

- ▶ Lignes directrices
 - ▶ *Cascading Style Sheet*
 - ▶ Séparer la structure du document de la présentation
 - ▶ Notion de feuille de style (mutualisée)
 - ▶ En cascade : cf héritage en POO
- ▶ Évolution en "niveaux"
 - ▶ CSS 1 1996
 - ▶ CSS 2 1998 - 2011
 - ▶ CSS 3 2011- modules : la référence actuelle
 - ▶ CSS 4 en développement
- ▶ Documentation
 - ▶ <http://www.w3.org/TR/html401/present/styles.html>
 - ▶ http://www.w3schools.com/cssref/css_selectors.asp

Appel CSS

- ▶ en ligne (*inline*) : déconseillé
- ▶ style en en-tête
déclaré dans un élément `<style>`, dans la section `<head>`
- ▶ feuilles de style externes
dans `<head>`, ligne d'inclusion
`<link rel="stylesheet" type="text/css"
href="theme.css" />`

Règles CSS

Composition d'une règle CSS

- ▶ un **sélecteur** CSS
 - ▶ un type d'élément ex. `<p>`
 - ▶ un attribut spécifique, ex.
 - ▶ `.citation` (attribut `class="citation"`)
 - ▶ `#MyId` (attribut `id="MyId"`)
 - ▶ un chemin DOM ex. `ul li a`
- ▶ des **propriétés**, par exemple :
 - ▶ `color: red;`
 - ▶ `font-weight: bold;`
 - ▶ `font-family: italic;`
 - ▶ `padding: 10 px 5 px 20 px 5 px;`
 - ▶ `display: hidden;`

Exemple

```
table, tr, td {  
    border: 1px solid black;  
}  
  
td {  
    padding: 5px;  
}  
  
.selected {  
    background-color: yellow;  
}
```

Le langage PHP