SI6

Json

Format & exploitation

JSON

* JavaScript Object Notation
* Utilise des étiquettes
* Peu verbeux
* Types de données élémentaires :
  * Tableaux : array [ ] = ensemble d'enregistrements
  * Objets : object { } = ensemble de couples (clés, valeurs)
    * Chaînes : string
    * Nombres : integer, …
    * Booléens : true, false

Outils de visualisation

* Parsers online : http://json.parser.online.fr
* Parsers applicatifs : JSON validator, …
* Editeurs de texte : c'est du texte

* Pourquoi ?
* Pour simplifier la lecture (validation, création de parser, …)

Parser

* C’est un analyseur syntaxique
* Repère la syntaxe (mot-clé, structure)

* JSON et XML utilisent une structure
* XML : balises XML <balise>valeur</balise>
* JSON : étiquettes {“étiquette”, “valeur”}

Exemple

* JSON

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

Exemple

* XML

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

Extension

* Il peut servir à faire de la sérialisation d'objets
* Encoder des fichiers
* Structurer des fichiers de configuration
* Implémenter dans 55 langages de programmation
* Utilise des bibliothèques externes
* Utilisé dans la programmation web
* Utile pour faire communiquer des applications hétérogènes

* JsonP (padding) est une amélioration de Json

Implémentations

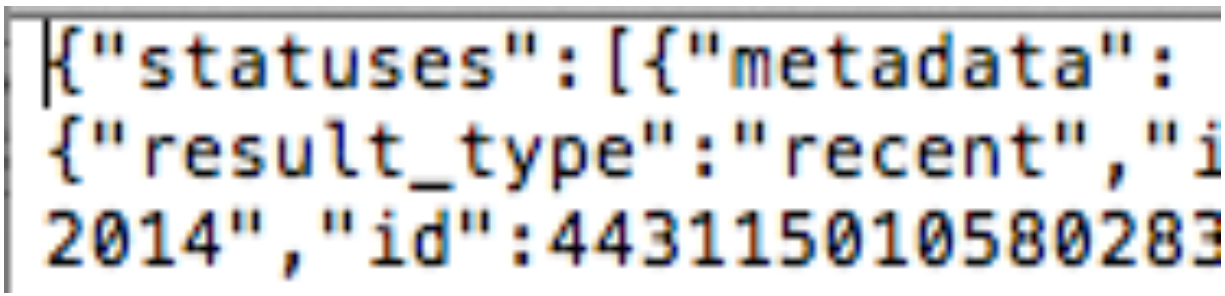
* Dans tous les langages des bibliothèques existes :
* Langage C : jansson
* Les implémentations rendent l'exploitation plus simple
* Sauf si vous souhaitez réinventer la roue !

Format d'échange

* Accès aux données de :
  * Twitter
  * Facebook
  * gitHub
* Via des APIrest (services restfull)
* Envoi des données en JSON
* Léger et rapide pour les périphériques portables

Notion de tableau

* Un tableau :



* json_t * searchData ;
* serachData = json_array() ;

Notion de tableau

* Tableau => boucle

```
searchData = json_object_get(json,"statuses");
if(json_is_array(searchData))
{
    for(i = 0; i < json_array_size(searchData); i++)
    {
        printf("********************************** TWEETS %d
```

Notion d'objet

* Objet dans un tableau :

* Json_t * data ;
* data = json_object() ;
* char * message_text ;

```
data = json_array_get(searchData, i);

//text
text = json_object_get(data, "text");
```

Types

* Extraction du type :

```
data = json_array_get(searchData, i);

//text
text = json_object_get(data, "text");
message_text = json_string_value(text);
printf("Message : %s \n", message_text);
```

Affichage

* Affichage de l'objet direct :

* json_dumps(json,0) ; // afficher l'objet sous la forme d'une chaîne %s
* json_dumpf(json,out,0) ; // afficher le fichier complet sous la forme d'une chaîne %s