

INF4375 - Paradigmes des échanges Internet

Formats de sérialisation

Jacques Berger

# Objectifs

Comprendre l'utilité d'un format de sérialisation

Introduire JSON, XML, HTML

# Prérequis

Aucun

# Sérialisation

Transformation d'une donnée en mémoire dans une représentation pouvant être communiquée au travers d'un réseau

La représentation peut être binaire ou texte

# Sérialisation

Déroulement :

Logiciel 1 sérialise une donnée

Logiciel 1 envoie la donnée sérialisée à Logiciel 2

Logiciel 2 reçoit la donnée sérialisée

Logiciel 2 désérialise la donnée

Logiciel 2 utilise la donnée

# Formats

Pour favoriser l'interopérabilité, il est nécessaire d'utiliser un format de sérialisation standardisé

Les formats de sérialisation standards sont habituellement des formats texte

# Formats

Formats connus :

JSON

XML

HTML

# JSON

JavaScript Object Notation

Sous-ensemble du langage Javascript



# JSON

Format texte

Interopérabilité

Simple et léger

Lisible par une machine et un humain

# JSON

Extension habituelle : .json

MIME Type : application/json

# JSON

```
{  
  "type": "book",  
  "title": "Théorie de la musique",  
  "author": [  
    "A. Danhauser"  
  ],  
  "year": 1996  
}
```

# Syntaxe

Aucun mot-clé réservé

Sensible à la casse

Concepts

Objets

Tableaux

# Objet

Un objet JSON est délimité par des accolades { }

Il contient des couples propriété/valeur

Le nom de la propriété est une chaîne de caractères

# Objet

La valeur peut être :

Une chaîne de caractères

Un nombre

true

false

null

Un objet

Un tableau

# Objet

La valeur est séparée de la propriété par un ":"

```
"type": "book"
```

Les couples sont séparés par une virgule

```
{  
  "type": "book",  
  "title": "Théorie de la musique"  
}
```

# Objet

```
{  
  "type": "book",  
  "title": "Test-Driven Development by Example",  
  "author": "Kent Beck",  
  "year": 2003  
}
```



# Tableau

## Array

Exactement comme un tableau en Javascript

Liste de valeurs ordonnées, sans nom

Le tableau est délimité par des crochets [ ]

Les valeurs sont séparées par une virgule

# Tableau

```
[  
  "Erich Gamma",  
  "Richard Helm",  
  "Ralph Johnson",  
  "John Vlissides"  
]
```

# Exemple complet

Voir exemple

# Encodage des données

Un document JSON est encodé en utf-8

# Vérification syntaxique

Vérifier la syntaxe un document JSON pour s'assurer de sa conformité est toujours une bonne idée

Outil en ligne : JSONLint  
<http://jsonlint.com/>

# XML

eXtensible Markup Language

Dernière version : 1.1

Couramment utilisée : 1.0

# XML

Norme du W3C (World Wide Web Consortium)

Création de langages

Balises

# Utilité

## Modéliser des données

Une exécution

Un algorithme

Une structure de données

Une configuration

Une image

Un graphique

Une formule mathématique

Une molécule chimique

Un document

Un livre

Un message

etc.



# Origine

## GML

Generalized Markup Language

Date des années 60

## SGML

Standard Generalized Markup Language

ISO 8879:1986

# Origine

## Les dérivés du SGML

HTML (avant la version 5)

XML

MathML

MusicML

etc.

# Abstraction

Un arbre syntaxique

Éléments auto-descriptifs

Contient de l'information

# Types de documents

Documents narratifs  
Sans schéma

Documents structuraux  
Avec schéma

# XML

Extension habituelle : .xml

MIME Type : application/xml

# Vocabulaire

## Document

Un fichier ou un ensemble de fichiers XML  
Les données, la structure, les instructions

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<document>
```

...

```
</document>
```

# Vocabulaire

## Élément

Un noeud de l'arbre (abstraction)

Peut contenir des données ou d'autres éléments

```
<element>  
  <contenu>texte</contenu>  
</element>
```

# Vocabulaire

## Élément vide

Un élément qui ne contient rien

```
<vide></vide>
```

```
<vide />
```



# Vocabulaire

## Élément racine

Le premier élément du document

L'élément qui contient tous les autres éléments

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<document>
```

...

```
</document>
```

# Vocabulaire

## Élément texte

Élément dont le contenu n'est que du texte

```
<element>Texte</element>
```

# Vocabulaire

## Élément enfant

Un élément à l'intérieur d'un autre élément

```
<parent>  
  <enfant>  
    ...  
  </enfant>  
</parent>
```

# Vocabulaire

## Attribut

Une donnée dans la déclaration d'un élément

```
<element attribut1="valeur1" attribut2="valeur2" />
```

# Vocabulaire

## Namespace

Une façon d'identifier qu'un élément appartient à un ensemble d'éléments spécifique

```
<xsl:template>
```

```
...
```

```
</xsl:template>
```

# Vocabulaire

## Racine du document

Un élément abstrait, non visible, situé avant l'élément racine

Ne pas confondre avec l'élément racine

# Vocabulaire

Noeud

Tout est un noeud

Élément

Commentaire

Instructions

Texte

# Règles de forme

Document bien formé

Respecte la norme du W3C

Document valide

Respecte son schéma

Note : un document bien formé et sans schéma est valide



# Règles de forme

## Un seul élément racine par document

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<document>  
...  
</document>
```

---

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<document>  
...  
</document>  
<document>  
...  
</document>
```

# Règles de forme

## Balise fermante obligatoire

```
<element>Texte</element>  
<element />
```

---

```
<element>  
  <enfant>  
</element>
```

# Règles de forme

Les identifiants sont sensibles à la casse

Éléments

Attributs

`<element> ≠ <Element>`

# Règles de forme

L'indentation est optionnelle

```
<document><element1><element2>Data</element2><element2>More  
Data</element2></element1></document>
```

# Règles de forme

Certains caractères spéciaux doivent être échappés

Exemple : '>' devient &gt;

Consultez la norme pour la liste complète

# Processing-instruction

Les documents XML commencent généralement par une ligne de ce genre :

```
<?xml version="1.0" encoding="utf-8"?>
```

Il s'agit d'une processing instruction

# Processing-instruction

La processing-instruction sert aux logiciels qui utiliseront ce document XML

Il s'agit de métadonnées pour le document

# Processing-instruction

On y spécifie :

- La version XML utilisée

- L'encodage de caractères des données

- Si le document doit être lié à un schéma

- Autres...



# Outils

Microsoft XML Notepad

XML Marker

XML Mind Editor

Eclipse, NetBeans et autres éditeurs

# XML et JSON

Il existe des outils de conversion automatique de XML vers JSON et vice versa

# XML et JSON

## Ce document XML :

```
<?xml version="1.0" encoding="utf-8"?>
<discs>
  <disc>
    <artist>Unearth</artist>
    <title>III: In the Eyes of Fire</title>
  </disc>
  <disc>
    <artist>Slipknot</artist>
    <title>Slipknot</title>
  </disc>
  <disc>
    <artist>Iced Earth</artist>
    <title>Alive in Athens</title>
  </disc>
</discs>
```

# XML et JSON

Deviendra :

```
{
  "discs": {
    "disc": [
      {
        "artist": "Unearth",
        "title": "III: In the Eyes of Fire"
      },
      {
        "artist": "Slipknot",
        "title": "Slipknot"
      },
      {
        "artist": "Iced Earth",
        "title": "Alive in Athens"
      }
    ]
  }
}
```

# HTML

HyperText Markup Language

Version actuelle : 5

# HTML

Le langage de base pour la création de sites web

Interprété par les fureteurs pour l'affichage des sites web

Sert à structurer le contenu d'un document web

# HTML

Extension habituelle : .html

MIME Type : text/html

# HTML

HTML est plutôt un format de publication web qu'un format de sérialisation

Par contre, des fragments HTML sont souvent utilisés comme format d'échange entre un serveur web et une application web



# HTML

Outre les échanges Ajax, on se sert rarement du format HTML pour échanger des données

Et même pour Ajax, l'utilisation du HTML pour échanger les données pourrait entraîner certains problèmes de sécurité

Malgré tout, ça demeure assez courant...

# Syntaxe

Langage balisé

Non sensible à la casse

Tous les caractères d'espacement sont remplacés  
par un seul espace

# Éléments de base

`<html>` : élément racine

`<head>` : contient les métadonnées du document

`<title>` : le titre du document

`<meta>` : l'attribut `charset` sert à spécifier l'encodage de caractères des données du document

# Éléments de base

`<body>` : le corps du document, contient les données

`<p>` : délimite un paragraphe

`<br>` : provoque un saut de ligne dans le texte

`<pre>` : texte préformaté, les caractères d'espacement sont préservés

# Éléments de base

`<table>` : crée un tableau

`<thead>` : Contient l'entête du tableau

`<tbody>` : Contient le corps du tableau

`<tfoot>` : Contient le bas du tableau

# Éléments de base

`<tr>` : crée une rangée dans un tableau

`<td>` : crée une cellule de données dans un tableau

`<th>` : crée une cellule d'en-tête dans un tableau

# Éléments de base

`<ul>` : crée une liste non ordonnée

`<ol>` : crée une liste ordonnée

`<li>` : crée un élément dans une liste

# Fonctionnalités

HTML comporte beaucoup d'autres fonctionnalités

Nous n'utiliserons que le strict minimum dans ce cours



# Validation

Pour s'assurer qu'on respecte la structure réelle d'un document HTML, il est important de valider nos documents

Les éditeurs et des outils en ligne nous permettent de le faire aisément

# Plus loin...

JSON

<http://www.json.org/>

JSONLint

<http://jsonlint.com/>

JavaScript : Gardez le meilleur!

Douglas Crockford

Annexe E

# Plus loin...

XML 1.0

<https://www.w3.org/TR/xml/>

XML 1.1

<https://www.w3.org/TR/xml11/>

HTML5

<http://www.w3.org/TR/html5/>

Validation HTML

<http://validator.w3.org/>