

Web linking

Chapitre 5
restful webservices cookbook
Cyprien Quilici

Liens en XML

- Atom fourni un format pour la syndication de contenu (principal concurrent de RSS)

```
<ressource xmlns:atom="http://www.w3.org/2005/Atom">  
  <atom:link link href="http://ducksarethebest.com/"  
    type="url"  
    rel="alternate"  
    title="Ducks are the best !"/>  
</ressource>
```

Liens en XML

- Les champs à remplir sont :
 - rel : type de lien
 - href : lien vers la ressource
 - title : nom lisible à afficher à l'écran
 - type : type de la ressource
 - hreflang : langue dans laquelle est la ressource
 - length : taille de la ressource

L'importance du champs rel

- Le champs rel définit :
 - Quelle ressource est pointée par le lien
 - Quel est le sens du lien
 - Quelles actions on peut effectuer sur sa cible
 - Quels sont les formats supportés pour les requêtes et réponse sur la ressource

Les pratiques à éviter

- Mettre une URL de base pour plusieurs liens

```
<recettes xml:base="http://www.marmittton.org">  
  <atom:link rel="http://www.marmittton.org/rels/recette"  
    href="/recette/1">  
  <atom:link rel="http://www.marmittton.org/rels/recette"  
    href="/recette/2">  
</recettes>
```

Liens en JSON

- De la même façon, seul le format change

```
{
  "link" : {
    "rel" : "alternate",
    "href" : "http://www.republiquedesmangues.fr/",
  }
}

{
  "links" : [
    {
      "rel" : "alternate",
      "href" : "http://www.republiquedesmangues.fr/"
    },
    {
      "rel" : "http://www.example.org/rels/sitebidon",
      "href" : "http://www.fallingfalling.com/",
    }
  ]
}
```

Liens dans les headers

- Il est possible de mettre des liens dans les headers http:
 - Le lien n'est pas dépendant d'un format
 - Pas besoin de lire le corps du payload

```
# Response
HTTP/1.1 200 OK
Content-Type: image/jpeg
Link: <http://img.over-blog.com/383x600/0/19/24/98/drole/pelican-urinoire-your-argument-is-invalid.jpg>;
rel="alternate;type="application/xml"
```

Les types de relations

- Il en existe déjà des standards

| Name | Purpose |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| self | Use this type to link to the preferred URI of the resource. |
| alternate | Use this type when providing a link to a URI for an alternative version of the same resource. |
| edit | Use this type to link to a URI that clients can use to edit the resource. |
| related | Use this type to link to a related resource. |
| previous and next | Use these types to link to the previous or next resource in an ordered series of resources. |
| first and last | Use these types to link to the first and last resources in an ordered series of resources, e.g., to the first and last resources in a collection. |

Les types de relations

- On peut en créer soi-même en :
 - Créant une documentation HTML
 - Donnant le lien vers cette documentation
 - Si le type est créé pour un usage public en le déclarant
- Exemple :

```
<atom:link rel="http://example.org/rels/banana"  
href="http://humour-fun.net/photos/banane-perverse-fraise-poire.jpg"/>
```

Les types de relations

- La doc doit comprendre :
 - Le but du type
 - Les types de cibles et de sources liées
 - Les méthodes http que l'on peut utiliser dessus
 - Les types de media attendus
- A retenir :
 - Les relations standard ne sont pas sensibles à la casse
 - Les relations privées le sont

=> Toujours utiliser des minuscules simplifie la vie

Des liens pour gérer le fil d'une application

| Méthode | URL | But |
|---------|-----------------------------------------------------------------------------------------------------|----------------------|
| POST | http://www.example.org/achat | Créer un client |
| POST | http://www.example.org/achat/{id}/articles | Choisir des articles |
| POST | http://www.example.org/achat/{id}/valider | Valider le panier |
| POST | http://www.example.org/achat/{id}/payer | Payer la commande |
| POST | http://www.example.org/achat/{id}/quitter | Ne pas payer |

Des liens pour gérer le fil d'une application

- Problèmes :
 - Le client doit savoir les règles de changement de page
 - Un client malveillant peut faire planter le processus
- Une solution :
 - Créer des liens qui n'apparaissent que quand les conditions sont remplies

Les liens éphémères

- Problème : comment faire des liens expirant ?
- Solution : un lien fournissant une relation qui indique la date de « péremption » du lien et ce qu'il faut faire s'il a expiré

```
<achat-req xmlns:atom="http://www.w3.org/2005/Atom">  
  <amount currency="USD">666</amount>  
  <atom:link rel="http://www.bananashop.org/purchase-req/auth"  
    href="http://www.bananashop.org/auth/ASBV_05_11_2013_1/22_02?_k=a1191fd35d23"/>  
</achat-req>
```

Gabarits d'URL

- Permettent au client d'ajouter de l'information dans l'URL

```
http://www.cougars.org/people?k={keyword}&a={age}&r={region}|
```

Gabarits d'URL

- Leur forme en XML et JSON

```
<link-template href="http://www.candys.org/client/{client-id}"  
               title="Visualiser les données clients"  
|rel="http://www.candys.org/rels/detail"/>
```

```
{  
  "link-template" : {  
    "rel" : "http://www.candys.org/rels/detail",  
    "href" : "http://www.example.org/client/{client-id}",  
    "title" : "Visualiser les données client"  
  }  
}
```

Utilisation des liens par le client

- Le client enregistre l'état des données et affiche les liens auxquels il a droit
- Si l'utilisateur se déconnecte puis se reconnecte, le client doit afficher les bons liens



Merci de votre attention

Si vous avez des questions, n'hésitez pas
à les poser !