Services webs

Olivier Caron

Polytech Lille
Avenue Paul Langevin Cité Scientifique Lille 1
59655 Villeneuve d'Ascq cedex

http://ocaron.plil.fr
Olivier.Caron@polytech-lille.fr

Services webs VS programmes webs

- Services et programmes webs sont accessibles via une URL

Services webs VS programmes webs

- Services et programmes webs sont accessibles via une URL
- Un programme web fournit en résultat une page HTML.

Services webs VS programmes webs

- Services et programmes webs sont accessibles via une URL
- Un programme web fournit en résultat une page HTML.
- Un service web fournit des données (JSON, XML, . . .) ou une section de code HTML

Services webs VS programmes webs

- Services et programmes webs sont accessibles via une URL
- Un programme web fournit en résultat une page HTML.
- Un service web fournit des données (JSON, XML, . . .) ou une section de code HTML
- Services et programmes peuvent avoir des paramètres (mode GET ou POST)

Services webs et PHP (1/4)

- Appel du service par une URL.

Services webs et PHP (1/4)

- Appel du service par une URL.
- Encodage des paramètres

Services webs et PHP (1/4)

- Appel du service par une URL.
- Encodage des paramètres
  - Tous les caractères ne sont pas autorisés (URL)

Services webs et PHP (1/4)

- Appel du service par une URL.
- Encodage des paramètres
  - Tous les caractères ne sont pas autorisés (URL)
  - Un encodage pour chaque caractère (ex: ' ' donne '%20')

Services webs et PHP (1/4)

- Appel du service par une URL.
- Encodage des paramètres
  - Tous les caractères ne sont pas autorisés (URL)
  - Un encodage pour chaque caractère (ex: ' ' donne '%20')
  - Encodage de paramètre :
    fonction PHP (une chaîne en résultat): `urlencode($chaine)`

Services webs et PHP (1/4)

- Appel du service par une URL.
- Encodage des paramètres
  - Tous les caractères ne sont pas autorisés (URL)
  - Un encodage pour chaque caractère (ex: ' ' donne '%20')
  - Encodage de paramètre :
    fonction PHP (une chaîne en résultat): `urlencode($chaine)`
  - Décodage de paramètre :
    fonction PHP (une chaîne en résultat): `urldecode($chaine)`

Services webs et PHP (2/4)

- Bibliothèque curl : invocation de services webs.

Services webs et PHP (2/4)

- Bibliothèque curl : invocation de services webs.
  - `$service=curl_init()` : initialisation

Services webs et PHP (2/4)

- Bibliothèque curl : invocation de services webs.
  - `$service=curl_init()` : initialisation
  - `curl_setopt($service, CONSTANTE_CURL, $param)` : configuration

Services webs et PHP (2/4)

- Bibliothèque curl : invocation de services webs.
  - `$service=curl_init()` : initialisation
  - `curl_setopt($service, CONSTANTE_CURL, $param)` : configuration
  - `curl_exec($service)` : invocation du service, récupération des données

Services webs et PHP (2/4)

- Bibliothèque curl : invocation de services webs.
  - `$service=curl_init()` : initialisation
  - `curl_setopt($service, CONSTANTE_CURL, $param)` : configuration
  - `curl_exec($service)` : invocation du service, récupération des données
  - `curl_close($service)` : fermeture

Services webs et PHP (3/4)

- Un exemple :

```
$url="http://sos-salle.polytech-lille.fr/serviceWhereIs.php";
$url.="?group=D400471/S4";                    // paramètre pour PEIP S4
$service = curl_init();                       // initialisation
curl_setopt($service, CURLOPT_URL, $url);     // configuration réseau
/** configuration réseau Polytech **/
curl_setopt($service, CURLOPT_PROXY, "proxy.polytech-lille.fr");
curl_setopt($service, CURLOPT_PROXYPORT, 3128);
/** fin configuration Polytech **/
//return the transfer as a string :
curl_setopt($service, CURLOPT_RETURNTRANSFER, 1);
$data = curl_exec($service);                  // invocation service
curl_close($service);                         // fermeture
echo $data ;                                  // affichage des données reçues
```

Services webs et PHP (4/4)

- Récupération des codes d'erreurs

```
...
$data = curl_exec($service);          // invocation service
$httpCode = curl_getinfo($service, CURLINFO_HTTP_CODE);
if ($httpCode == 404) {
  echo "page web inexistante...";
  exit();
}
...
```

Echange de données

- Format de données XML :

Echange de données

- Format de données XML :
  - Avantages : langage structuré, format texte.

Echange de données

- Format de données XML :
  - Avantages : langage structuré, format texte.
  - Inconvénients : verbeux, lourd à décoder

Echange de données

- Format de données XML :
  - Avantages : langage structuré, format texte.
  - Inconvénients : verbeux, lourd à décoder
- Format de données JSON (JavaScript Object Notation) :

Echange de données

- Format de données XML :
  - Avantages : langage structuré, format texte.
  - Inconvénients : verbeux, lourd à décoder
- Format de données JSON (JavaScript Object Notation) :
  - format de données textuel, générique

Echange de données

- Format de données XML :
  - Avantages : langage structuré, format texte.
  - Inconvénients : verbeux, lourd à décoder
- Format de données JSON (JavaScript Object Notation) :
  - format de données textuel, générique
  - Avantages : simplicité, décodage rapide, typage des données

Structure de documents JSON

- Un document JSON ne comprend que deux éléments structurels :

Structure de documents JSON

- Un document JSON ne comprend que deux éléments structurels :
  - des ensembles de paires nom / valeur

Structure de documents JSON

- Un document JSON ne comprend que deux éléments structurels :
  - des ensembles de paires nom / valeur
  - des listes ordonnées de valeurs

Structure de documents JSON

- Un document JSON ne comprend que deux éléments structurels :
  - des ensembles de paires nom / valeur
  - des listes ordonnées de valeurs
- Ces mêmes éléments représentent 3 types de données :

Structure de documents JSON

- Un document JSON ne comprend que deux éléments structurels :
  - des ensembles de paires nom / valeur
  - des listes ordonnées de valeurs
- Ces mêmes éléments représentent 3 types de données :
  - des objets ('{' . . . '}')

Structure de documents JSON

- Un document JSON ne comprend que deux éléments structurels :
  - des ensembles de paires nom / valeur
  - des listes ordonnées de valeurs
- Ces mêmes éléments représentent 3 types de données :
  - des objets ('{' . . . '}')
  - des tableaux ('[' . . . ']')

Structure de documents JSON

- Un document JSON ne comprend que deux éléments structurels :
  - des ensembles de paires nom / valeur
  - des listes ordonnées de valeurs
- Ces mêmes éléments représentent 3 types de données :
  - des objets ('{' . . . '}')
  - des tableaux ('[' . . . ']')
  - des valeurs génériques de type tableau, objet, booléen (true, false) , nombre, chaîne ou null.

Un exemple JSON

```
{
  "name": "Frank",
  "age": 24,
  "engaged": true,
  "favorite_tv_shows": [
      "Lost", "Dirty_Jobs",
      "Deadliest_Catch", "Man_vs_Wild"
                          ]
}
```

PHP et JSON

- Utilisation des tableaux associatifs PHP
- Deux fonctions : `json_encode($tab)` et `json_decode($tab)`

```
$tab=array (
 "name" => "Franck", "age" => 24, "engaged" => true,
 "favorite_tv_shows" => array ("Lost", "Dirty Jobs" ,
                               "Deadliest catch", "Man vs Wild")
) ;
$chaineJSON=json_encode($tab);
```

Poursuite de l'exemple "sos-salle"

- Un exemple :

```
$url="http://sos-salle.polytech-lille.fr/serviceWhereIs.php";
$url.="?group=D400471/S4";
...
$data = curl_exec($service);                    // invocation service
curl_close($service);                           // fermeture
$objetJSON=json_decode($data);
echo "Liste des salles";
echo "<ul>";
foreach($objetJSON->current as $salle)
  echo "<li>$salle->room</li>";
echo "</ul>";
```

- PHP: accès à un objet ({ "name":"Franck" ...}) : `$var->name`
- PHP: accès à un élément de tableau ({ "favorite_tv_shows": [...] }) : `$var->favorite_tv_shows[2]`

Le service de géolocalisation de Google

- on fournit une adresse texte, on obtient les coordonnées latitude et longitude.

Le service de géolocalisation de Google

- on fournit une adresse texte, on obtient les coordonnées latitude et longitude.
- Adresse :

  `https://maps.googleapis.com/maps/api/geocode/json`

Le service de géolocalisation de Google

- on fournit une adresse texte, on obtient les coordonnées latitude et longitude.
- Adresse :
  `https://maps.googleapis.com/maps/api/geocode/json`
- paramètre `address`: l'adresse texte

Le service de géolocalisation de Google

- on fournit une adresse texte, on obtient les coordonnées latitude et longitude.
- Adresse :
  `https://maps.googleapis.com/maps/api/geocode/json`
- paramètre `address`: l'adresse texte
- key : une clé d'accès au service

Le service de géolocalisation de Google

- on fournit une adresse texte, on obtient les coordonnées latitude et longitude.
- Adresse : `https://maps.googleapis.com/maps/api/geocode/json`
- paramètre `address`: l'adresse texte
- key : une clé d'accès au service
- Pour obtenir une clé : `https://developers.google.com/maps/documentation/javascript/get-api-key`

Un exemple

- `address` : 10 rue de Turenne, Lille


```
{
{"results" : [
  {
    "address_components" : [ ... ],
    "formatted_address" : "10 Rue de Turenne, 59000 Lille, France",
    "geometry" : {
       "location" : {
          "lat" : 50.6292803,
          "lng" : 3.0359424
       }, ...
  }],
  "status" : "OK"
}
```


- récupérer latitude en PHP :

```
$obj->results[0]->geometry->location->lat
```