

Bachelier en Informatique de Gestion

Web : principes de base Projet de Développement Web

Enseignement supérieur économique de type court

Code FWB : 7534 29 U32 D1, 7534 30 U32 D3

Code ISFCE : 4IWPB, 4IPW3



Table des matières

Généralités

- 01. Introduction au web
- 03. Outils
- 05. Format XML
- 06. Format JSON

Front-End

- 12. Structure HTML
- 13. Formulaire HTML
- 14. Mise en forme CSS
- 15. Adaptabilité
- 17. Javascript
- 18. Bibliothèque jQuery
- 19. Composant Vue.js

Back-End

- 21. Middleware PHP
- 22. Traitement du formulaire
- 23. Architecture MVC
- 24. Données SQL
- 25. Données NoSQL
- 27. Requête asynchrone



= BUROTIX 0

26. Intégration de données NoSQL

intégration XML

intégration JSON

API



= BUROTIX 0

XML : intégration en PHP



= BUROTIX 0

Exo 11 : PHP et XML

- Données XML fournies sous forme de `string` php.
- Fonction PHP
`simplexml_load_string()`
- Exemple
 - `exo11.php`

```
$string = <<< XML
<?xml version='1.0'?>
<document>
    <cmd>login</cmd>
    <login>Richard</login>
    <prop>
        <admin>0</admin>
        <restricted_user>1</restricted_user>
    </prop>
</document>
XML;
```



simplexml_load_string()

- Pour convertir un string XML en un **objet PHP**
- Référence
 - <https://www.php.net/manual/fr/function.simplexml-load-string.php>



Exo 12 : PHP et XML

- Données XML : fichier local

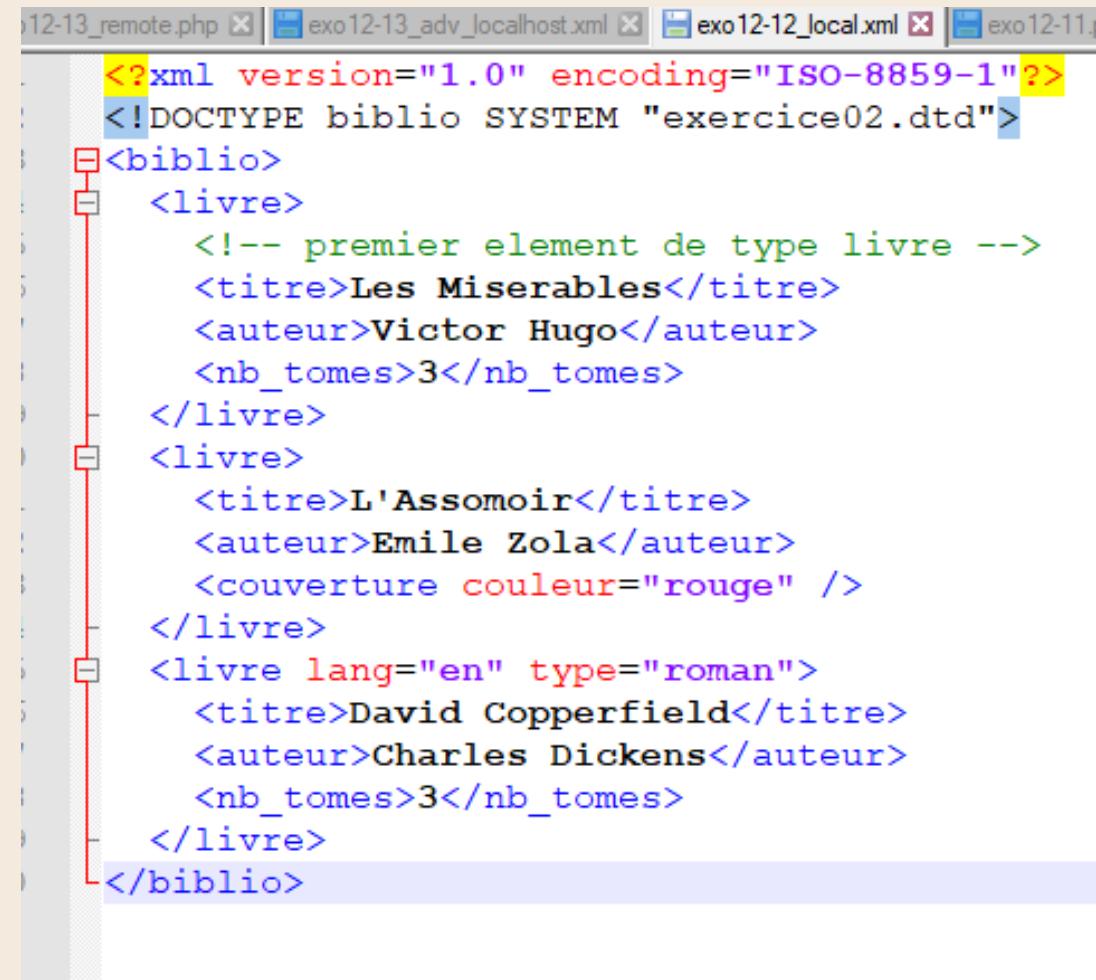
- Fonction PHP

`simplexml_load_file()`

- Exemple

`library+dtd+xsl.xml`

`exo12_local.php`



The screenshot shows a code editor window with several tabs at the top: 'exo12-13_remote.php', 'exo12-13_adv_localhost.xml', 'exo12-12_local.xml', and 'exo12-11'. The main area displays the following XML code:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE biblio SYSTEM "exercice02.dtd">
<biblio>
  <livre>
    <!-- premier element de type livre -->
    <titre>Les Miserables</titre>
    <auteur>Victor Hugo</auteur>
    <nb_tomes>3</nb_tomes>
  </livre>
  <livre>
    <titre>L'Assomoir</titre>
    <auteur>Emile Zola</auteur>
    <couverture couleur="rouge" />
  </livre>
  <livre lang="en" type="roman">
    <titre>David Copperfield</titre>
    <auteur>Charles Dickens</auteur>
    <nb_tomes>3</nb_tomes>
  </livre>
</biblio>
```



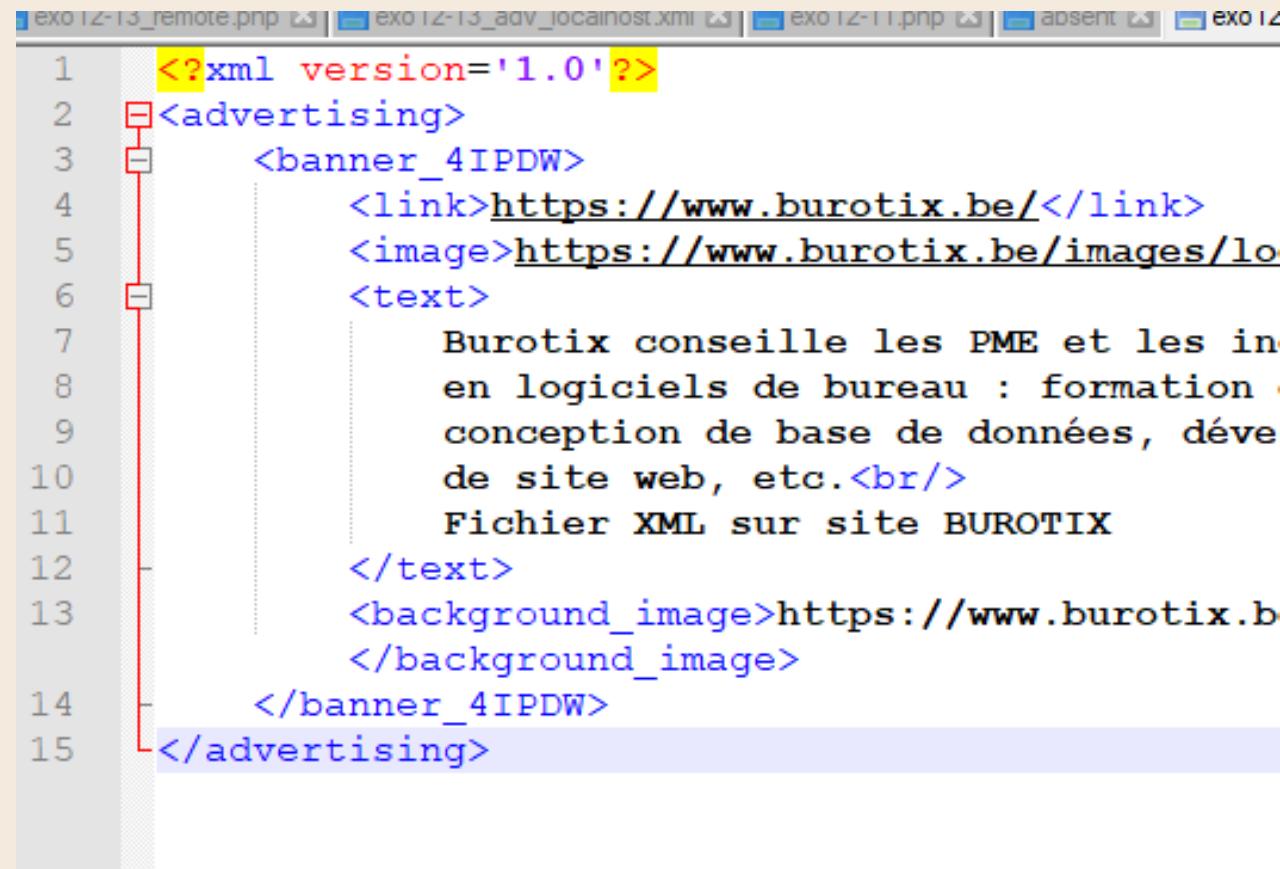
simplexml_load_file()

- Pour convertir un fichier XML en objet PHP
- Fichier XML étant
 - Soit **local**, en donnant le "path"
 - Soit **remote**, en donnant l' "URL" !



Exo 13 : PHP et XML

- Données XML : localisées sur un serveur
- Fonction PHP `simplexml_load_file()`
- Exemple
`exo13_remote.php`



The screenshot shows a code editor with several tabs at the top: "exo13_remote.php", "exo13_adv_localhost.xml", "exo13-11.php", "absent", and "exo13". The main pane displays the following XML code:

```
<?xml version='1.0'?>
<advertising>
  <banner_4IPDW>
    <link>https://www.burotix.be/</link>
    <image>https://www.burotix.be/images/lo</image>
    <text>
      Burotix conseille les PME et les in
      en logiciels de bureau : formation
      conception de base de données, déve
      de site web, etc.<br/>
      Fichier XML sur site BUROTIX
    </text>
    <background_image>https://www.burotix.b</background_image>
  </banner_4IPDW>
</advertising>
```



= BUROTIX 0

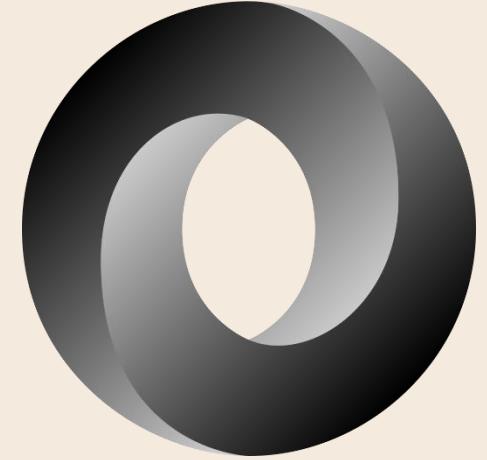
Exo 18 : PHP et XML

- BONUS : La table de conversion des devises offerte par la BCE
 - http://www.ecb.europa.eu/stats/eurofxref/eur_ofxref-daily.xml
 - Exemple :
exo18_currency.php

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<gesmes:Envelope xmlns:gesmes="http://www.gesmes.org/xml/2002-08-01"
  xmlns="http://www.ecb.int/vocabulary/2002-08-01/eurofxref">
  <gesmes:subject>Reference rates</gesmes:subject>
  <gesmes:Sender>
    <gesmes:name>European Central Bank</gesmes:name>
  </gesmes:Sender>
  <Cube>
    <Cube time="2020-03-05">
      <Cube currency="USD" rate="1.1187"/>
      <Cube currency="JPY" rate="119.63"/>
      <Cube currency="BGN" rate="1.9558"/>
      <Cube currency="CZK" rate="25.346"/>
      <Cube currency="DKK" rate="7.4712"/>
      <Cube currency="GBP" rate="0.86670"/>
      <Cube currency="HUF" rate="335.70"/>
      <Cube currency="PLN" rate="4.3029"/>
      <Cube currency="RON" rate="4.8110"/>
      <Cube currency="SEK" rate="10.5915"/>
      <Cube currency="CHF" rate="1.0663"/>
      <Cube currency="ISK" rate="142.20"/>
      <Cube currency="NOK" rate="10.3710"/>
      <Cube currency="HRK" rate="7.4905"/>
      <Cube currency="RUB" rate="74.3070"/>
      <Cube currency="TRY" rate="6.8038"/>
      <Cube currency="AUD" rate="1.6917"/>
      <Cube currency="BRL" rate="5.1480"/>
      <Cube currency="CAD" rate="1.5021"/>
      <Cube currency="CNY" rate="7.7579"/>
      <Cube currency="HKD" rate="8.6936"/>
      <Cube currency="IDR" rate="15851.98"/>
      <Cube currency="ILS" rate="3.8814"/>
      <Cube currency="INR" rate="82.1530"/>
      <Cube currency="MXN" rate="20.0550"/>
    </Cube>
  </Cube>
</gesmes:Envelope>
```





JSON : intégration en PHP



= BUROTIX 0

Exo 21 : PHP et JSON

- Données JSON fournies sous forme de string php.
- Fonction PHP
json_decode()
- Exemple
 - **exo21.php**

```
$string = <<< JSON
{
    "id": "Margaret",
    "image": "./model/media/robes.jpg",
    "type": "Robe",
    "ref": "41V2138",
    "couleur": "Virtual Pink",
    "lien": "http://www.desigual.com/fr_BE",
    "prix": 79,
    "taille": [
        "XL",
        "L"
    ]
}
JSON;
```



json_decode()

- Pour convertir un string JSON en un **objet** ou un **assoc array** PHP

```
$array = json_decode($string, $bool);
```

- Params
 - **\$string** : la chaîne sous format JSON
 - **\$bool** : **false** => return object (default)
true => return assoc array

- Référence
 - <https://www.php.net/manual/fr/function.json-decode.php>



Exo 23 : PHP et JSON

- Données JSON : localisées sur un serveur
- Fonctions PHP

file_get_contents()
json_decode()

- Exemple

exo23.php



A screenshot of a web browser window. The address bar shows the URL `playground.burotix.be/adv/banner_for_iffce.json`. The page content displays a JSON object:

```
{  
  "banner_4IPDW": {  
    "link": "https://www.burotix.be/",  
    "image": "https://www.burotix.be/images/logo103x90.png",  
    "text": "Entrepreneur, ind\u00e9pendant, artisan, ...  
sise \u00e0 Walhain dans le Brabant Wallon, optimise l'emploi  
\n\\t\\t\\ttableurs, bases de donn\u00e9es, sites web, dossier...  
etc.",  
    "background_image": "https://www.burotix.be/images/516277_480.jpg",  
    "color": "#0dd3d1"  
  }  
}
```

Exo 24 : PHP et JSON

- Données JSON : produites par votre serveur (export)
- Fonctions PHP :

json_encode()

- Exemple :

exo24.php



A screenshot of a web browser window displaying a JSON object. The URL in the address bar is `playground.burotix.be/adv/banner_for_iffce.json`. The JSON content is as follows:

```
{  
    "banner_4IPDW": {  
        "link": "https://www.burotix.be/",  
        "image": "https://www.burotix.be/images/logo103x90.png",  
        "text": "Entrepreneur, ind\u00e9pendant, artisan, ...  
sise \u00e0 Walhain dans le Brabant Wallon, optimise l'emploi  
\n\\t\\t\\ttableurs, bases de donn\u00e9es, sites web, dossier  
etc.",  
        "background_image": "https://www.burotix.be/images/516277_480.jpg",  
        "color": "#0dd3d1"  
    }  
}
```

json_encode()

- Pour convertir un objet ou un assoc array PHP en un string JSON

```
$json_str = json_encode( $array, $flags );
```

- Params

- **\$array** : les données sous format object ou assoc array ou ...
- **\$flags** :

JSON_HEX_TAG

JSON_PRETTY_PRINT

JSON_FORCE_OBJECT etc.

- Référence

- <https://www.php.net/manual/fr/function.json-encode.php>
- <https://www.php.net/manual/fr/json.constants.php>



A screenshot of a web browser window displaying a JSON object. The address bar shows the URL: localhost/13_json/exo13-14_make_json.php. The JSON content is:

```
{  
    "id": "Margaret",  
    "image": "./model/media/robes.jpg",  
    "type": "Robe",  
    "ref": "41V2138",  
    "couleur": "Virtual Pink",  
    "lien": "http://www.desigual.com/fr_BE",  
    "prix": 79,  
    "taille": [  
        "XL",  
        "L"  
    ]  
}
```

API (REST)



= BUROTIX 0

API : Principe

- Application Programming Interface
- Ensemble de définitions et de protocoles qui permettent à un logiciel de communiquer avec un autre logiciel
- Architecture client-serveur
- Scénario
 - L'application cliente envoie une demande à l'application serveur via une API.
 - L'application serveur répond ensuite à la demande de l'application cliente en renvoyant les données demandées.
 - format des données : JSON, XML, ou CSV.



API : Applications

- récupération de données
- envoi de données
- automatisation de tâches
- ajouter des fonctionnalités à une application
- améliorer l'expérience utilisateur
 - Par exemple, une application de planification de voyage peut utiliser l'API Google Maps pour afficher une carte et les directions vers une destination.



API : Types

- API REST
 - protocoles HTTP, HTTPS
 - méthodes GET, POST
- API SOAP
 - via XML
 - plus rigide et plus formel que REST
- API de bibliothèque
 - API internes permettant à un logiciel de communiquer avec les bibliothèques système.
 - API modernes:
 - GraphQL (Facebook)
 - gRPC (Google)
 - Websockets
 - Webhooks



Exo 33 : log-in, log-out avec API et JSON

- Corollaire à chap. 22, exo 24.
 - mécanisme de log-in
 - nom et mot de passe
 - identification permanente :
`$_SESSION`
- Mais
 - Validation sur un serveur extérieur
- Envoi des données
 - GET : simple
 - POST : sécurisé
- Retour : format JSON
 - {
 "identified": true,
 "name": "Luke Skywalker",
 "role": "user"
}



Exo 33 : log-in, log-out avec API et JSON

- Envoi des données GET : `validate_get()`
 - construire un URL tel que
`http://...php?login=user&passwd=user`
avec `http_build_query()`
 - appeler `file_get_contents($uri)`
 - traiter le retour



Exo 33 : log-in, log-out avec API et JSON

- Envoi des données POST : `validate_post()`
 - URL tel que `https://....php` (sans param)
 - URL param : idem GET
 - créer un "contexte" tel que

```
"http" => [
    "method" => "POST",
    "header" => "Content-Type: ...".
    "content" => $uri_param_s,
]
```
 - appeler `file_get_contents($uri, false, $context)`
 - traiter le retour

