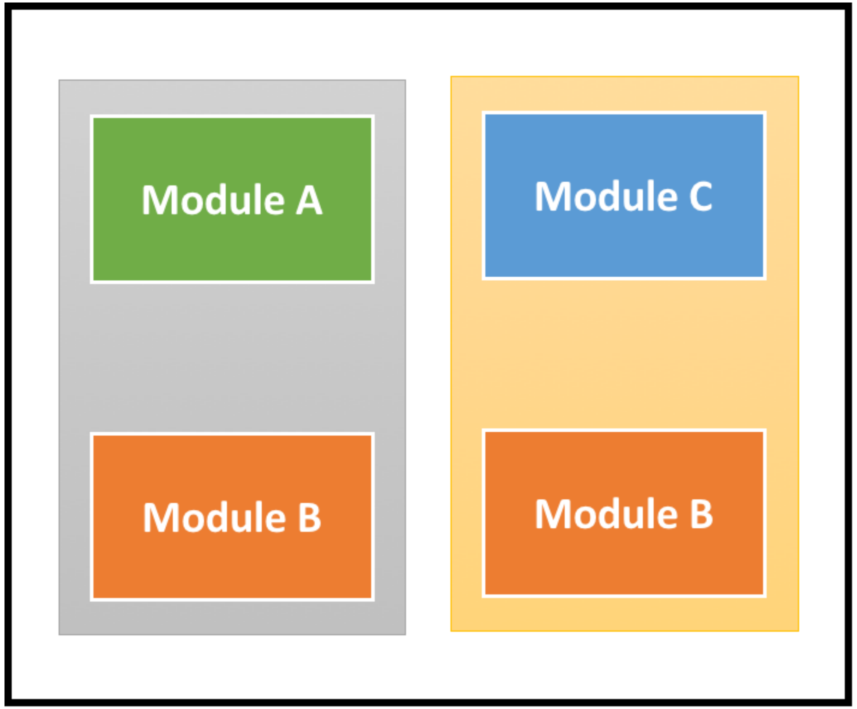
Web Services Programming

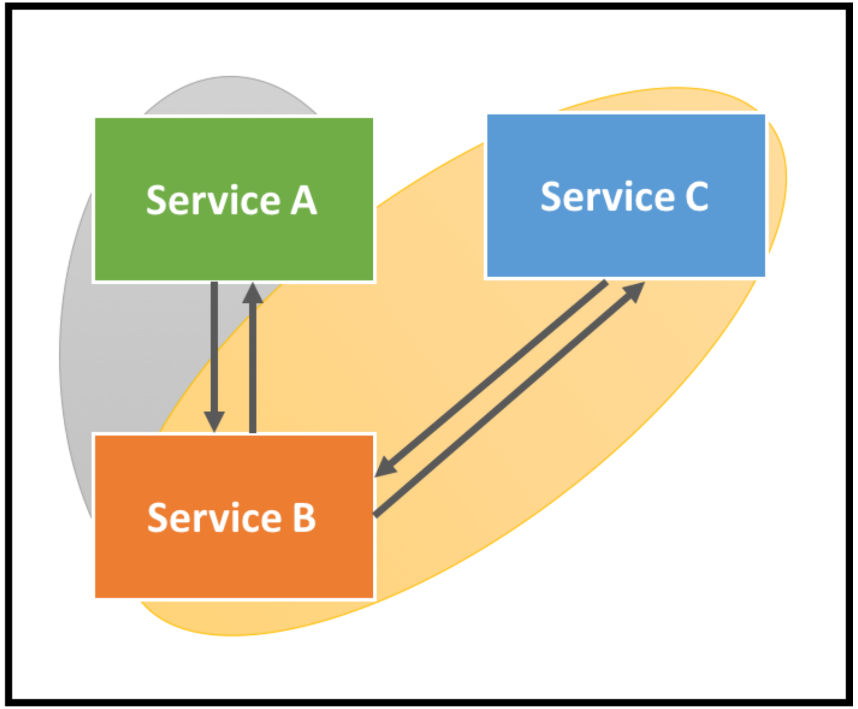
DWES UD7.1

Index

1.- Introduction

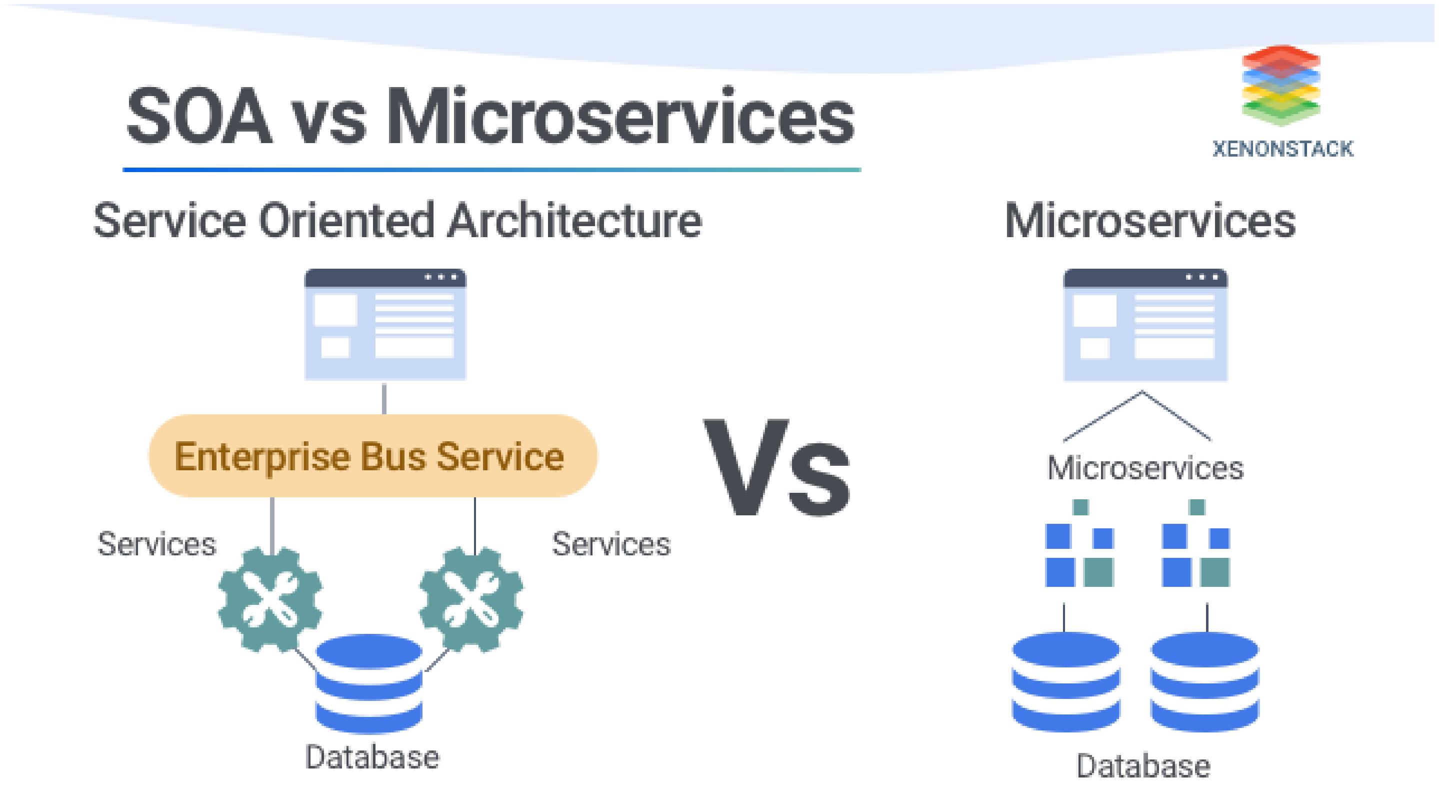


Monolithic Architecture



Service-Oriented Architecture

1.- Introduction



1.- Introduction

Standards used:

- HTML
- XML
- JSON
- SOAP (Simple Object Access Protocol)
- WSDL (Web Services Description Language)
- RESTFUL (REpresentational State Transfer)
- UDDI (Universal Description, Discovery and Integration)

2.- Reading XML files

There are many methods to parse XML:
https://www.php.net/manual/en/refs.xml.php

They are grouped into two types:

- Tree-based parsers: Simple XML and dom
- Event-Based Parsers: XML Reader

We are going to use SimpleXML.

2.- Reading XML files – Simple XML

- A string can be read with the function simplexml_load_string()

```
<?php
$string = <<<XML
<?xml version='1.0'?>
<document>
 <title>Forty What?</title>
 <from>Joe</from>
 <to>Jane</to>
 <body> I know that's the answer -- but what's the question? </body>
</document>
XML;

$xml = simplexml_load_string($string);
print_r($xml);
?>
```

2.- Reading XML files – Simple XML

- The result of the previous example will show the following:

```
SimpleXMLElement Object
(
  [title] => Forty What?
  [from] => Joe
  [to] => Jane
  [body] =>
   I know that's the answer -- but what's the question?
)
```

2.- Reading XML files – Simple XML

- A file can be read with the function simplexml_load_file()

- If we have the 'note.xml' file with the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

2.- Reading XML files – Simple XML

- Information can be displayed as follows:

```
<?php
   $xml=simplexml_load_file("note.xml");
   echo $xml->to . "<br>";
   echo $xml->from . "<br>";
   echo $xml->heading . "<br>";
   echo $xml->body;
?>
```

- The previous script would return:

```
Tove
Jani
Reminder
Don't forget me this weekend!
```

2.- Reading XML files – Simple XML

Using the 'books.xml' file from aules. The following example obtains the value of the 'category' attribute of the first book and the value of the language attribute of the 'title' element of the second book:

```
<?php
        $xml=simplexml_load_file("books.xml");
        echo $xml->book[0]['category'] . "<br>";
        echo $xml->book[1]->title['lang'];
?>
```

The previous script would return:

```
COOKING
en
```

2.- Reading XML files – Simple XML

What would the following example return?:

```
<?php
        $xml=simplexml_load_file("books.xml");
        foreach($xml->children() as $books) {
          echo $books->title['lang'];
          echo "<br>";
        }
?>
```

2.- Reading XML files – Simple XML

You can also navigate the xml file by iterating with the functions:

FUNCTION	DESCRIPTION
current()	Returns the current element
getChildren()	Returns the children elements of the current one
hasChildren()	Checks if the current element has children
key()	Returns the XML tag of the current element
next()	Select the next item
rewind()	Go back to the first item
valid()	Checks if the current item is valid

More examples:
https://www.php.net/manual/en/simplexml.examples-basic.php

Exercise

- Download the books.xml file and try all previous examples
- Create a script that searches for books in the web category, then shows their title and price

3.- Reading JSON files

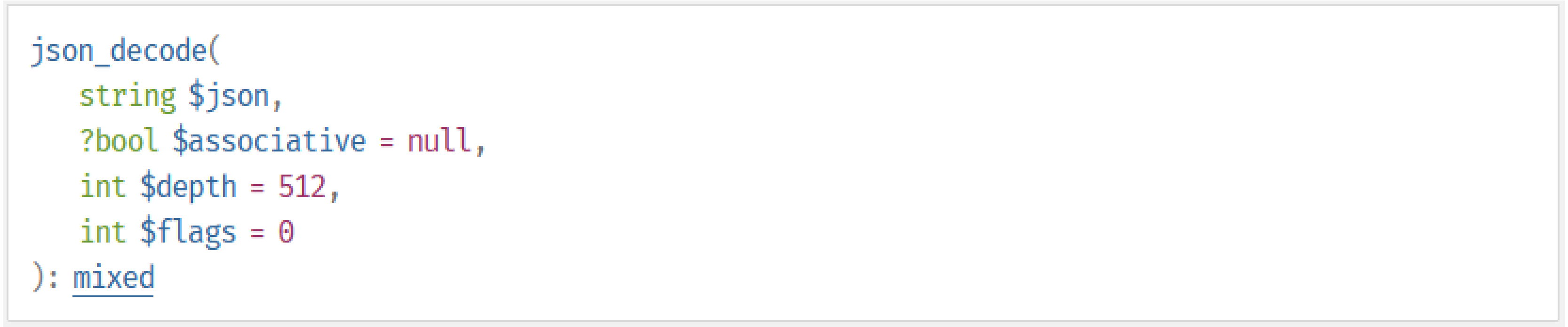
There are mainly two ways to read json:

- Json_decode
- Unserialize

{JSON}
JavaScript Object Notation

The recommended method that we are going to see is json_decode

3.- Reading JSON files



```
json_decode(
    string $json,
    ?bool $associative = null,
    int $depth = 512,
    int $flags = 0
): mixed
```

If you run the following script:

```
<?php
   $json = '{"a":1,"b":2,"c":3,"d":4,"e":5}';

   var_dump(json_decode($json));
   var_dump(json_decode($json, true));
?>
```

3.- Reading JSON files

```
object(stdClass)#1 (5) {
    ["a"] => int(1)
    ["b"] => int(2)
    ["c"] => int(3)
    ["d"] => int(4)
    ["e"] => int(5)
}

array(5) {
    ["a"] => int(1)
    ["b"] => int(2)
    ["c"] => int(3)
    ["d"] => int(4)
    ["e"] => int(5)
}
```

3.- Reading JSON files

The elements in the array can be consulted like this:

```
<?php
	$jsonobj = '{"Peter":35,"Ben":37,"Joe":43}';
	$arr = json_decode($jsonobj, true);
	echo $arr["Peter"];
	echo $arr["Ben"];
	echo $arr["Joe"];
?>

<?php
	$json = '{"foo-bar": 12345}';
	$obj = json_decode($json);
	print $obj->{'foo-bar'}; // 12345
?>
```

3.- Reading JSON files

A json file can be read using file_get_contents:

```
<?php
        $jsonobj = file_get_contents(‘file_input.json’);
        print_r($jsonobj);
?>
```

Exercise

- Download the marvel_characters.json file
- Create a script that shows in table format the name of all characters with the name of all the comics where they appear

4.- Access to RESTful services

REST (Representational State Transfer) is an "architectural style" that basically takes advantage of existing Web technology and protocols. **RESTful** is often used to refer to web services running such an architecture.

The response from a restful service can be XML or JSON.

There are several ways to access restful services, we are going to see two:

- file_get_contents()
- cURL

4.- Access to RESTful services (JSON)

With file_get_contents you can access the service, decode it in a simple way and obtain an array object with which you can work:

```
<?php
		$json_data =
file_get_contents('https://jsonplaceholder.typicode.com/posts/');
		$json_decod = json_decode($json_data);
		var_dump($json_decod);
?>
```

4.- Access to RESTful services (JSON)

cURL is a little more complicated but it is very powerful since it has many options:

```
<?php
        $ch = curl_init(); //se inicia sesión
        $url = 'https://jsonplaceholder.typicode.com/posts/';
        curl_setopt($ch, CURLOPT_URL,$url); //se configura la url
        $json_data = curl_exec($ch); //se accede al servicio
        curl_close($ch); //se cierra sesión
        $json_decod = json_decode($json_data);
        var_dump($json_decod);
?>
```

4.- Access to RESTful services (JSON)

An example of using options: https://www.php.net/manual/en/curl.constants.php#76914

```
...
  curl_setopt($curl, CURLOPT_URL, $url);
  curl_setopt($curl, CURLOPT_USERAGENT, 'Googlebot/2.1 (+http://www.google.com/bot.html)');
  curl_setopt($curl, CURLOPT_HTTPHEADER, $header);
  curl_setopt($curl, CURLOPT_REFERER, 'http://www.google.com');
  curl_setopt($curl, CURLOPT_ENCODING, 'gzip,deflate');
  curl_setopt($curl, CURLOPT_AUTOREFERER, true);
  curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
  curl_setopt($curl, CURLOPT_TIMEOUT, 10);
...
```

4.- Access to RESTful services (XML)

With file_get_contents you access the service and convert the information with one of the methods we have already seen:

```
<?php
    $respuesta =
file_get_contents('https://www.aemet.es/xml/municipios_h/localidad_h_46244.xml’);
    $datos = simplexml_load_string($respuesta);
    echo "<pre>"; print_r($datos);
?>
```

4.- Access to RESTful services (XML)

With cURL it could be done as follows:

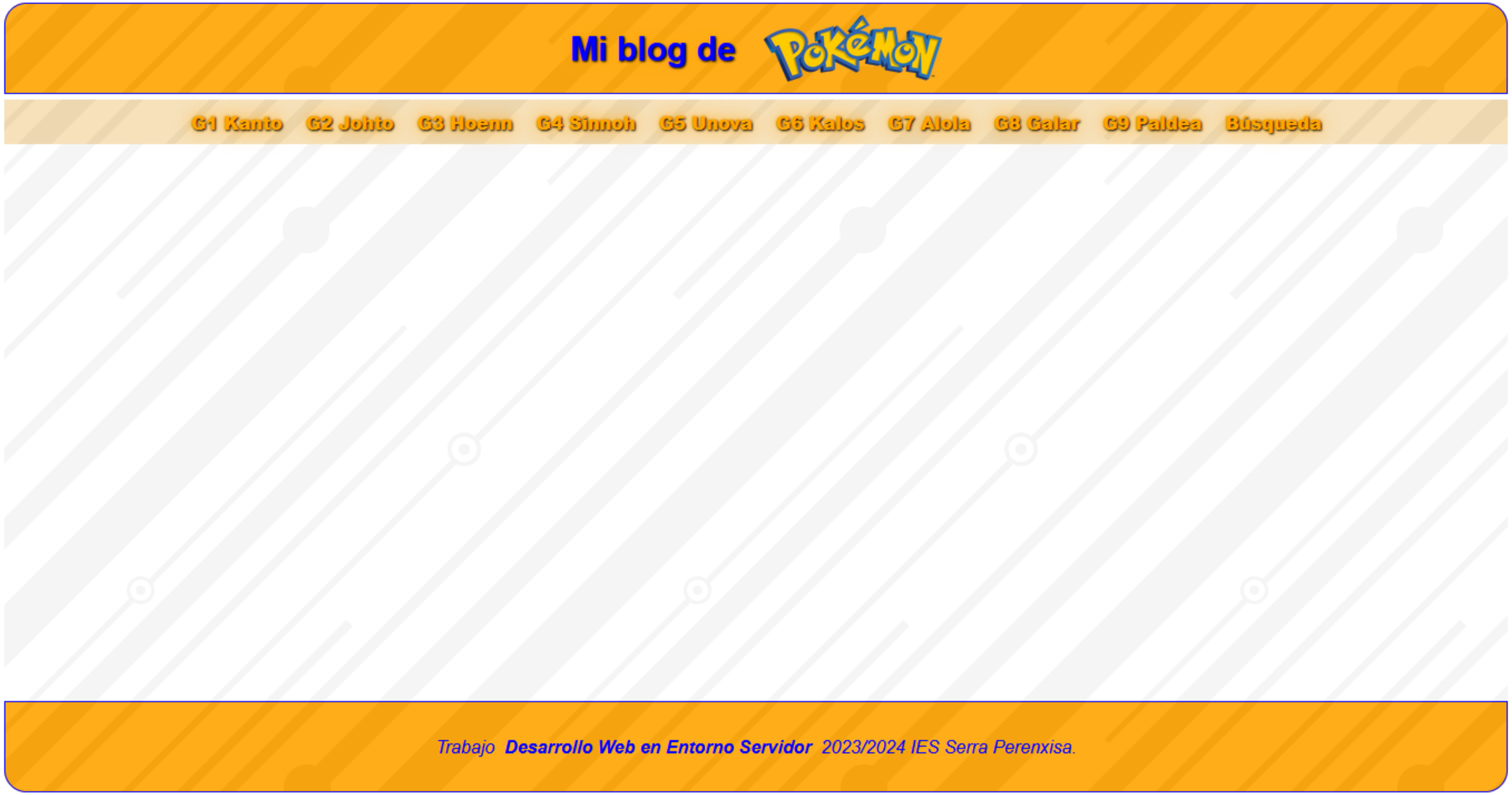
```
<?php
  $ch = curl_init(); //se inicia sesión
  $url = 'https://www.aemet.es/xml/municipios_h/localidad_h_46244.xml’;
  curl_setopt($ch, CURLOPT_URL,$url); //se configura la url
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
  $respuesta = curl_exec($ch); //se accede al servicio
  curl_close($ch); //se cierra sesión
  $datos = simplexml_load_string($respuesta);
  echo "<pre>"; print_r($datos);
?>
```

Exercise

Create an application that shows Pokemon's information using the Pokeapi:

- The documentation is here
- The main api endpoint is https://pokeapi.co/api/v2/
- The regions endpoint is https://pokeapi.co/api/v2/region/
- The pokemons endpoint is https://pokeapi.co/api/v2/pokemon/
- The 'next' field of json response contains the following elements
- The application main page must show all regions
- When the user selects a region, it will show the pokemons in that region
- When the user access a pokemon it will show the details with an image
- Include a search page that allows you to search by name, region and type
- The application must have a menu in all pages

Exercise – Main Page



Mi blog de

G1 Kanto G2 Johto G3 Hoenn G4 Sinnoh G5 Unova G6 Kalos G7 Alola G8 Galar G9 Paldea Búsqueda

Trabajo **Desarrollo Web en Entorno Servidor** 2023/2024 IES Serra Perenxisa.

How to read from a file

Assuming we have the file ‘credenciales.txt’ with the following content:

```
{"host":"mihost","username":"miusuario","password":“mipassword","database":"mibasededatos"}
```

This script opens the file and reads only one line:

```php
<?php
    $ficheroCredenciales = fopen('credenciales.txt', 'r');
    $credenciales = json_decode(fgets($ficheroCredenciales), true);
    var_dump($credenciales);
    fclose($ficheroCredenciales);
?>
```

Will output:

```
array(4) { ["host"]=> string(6) "mihost" ["username"]=> string(9) "miusuario" ["password"]=>
string(10) “mipassword" ["database"]=> string(13) "mibasededatos" }
```

How to read from a file

Assuming we have the file ‘credenciales.txt’ with the following content:

```
{"host":"mihost","username":"miusuario","password":“mipassword","database":"mibasededatos"}
```

This script reads the whole file and returns it as a string:

```
<?php
   $credenciales = json_decode(file_get_contents('credenciales.txt’));
   var_dump($credenciales);
?>
```

Will output:

```
object(stdClass)#1 (4) { ["host"]=> string(6) "mihost" ["username"]=> string(9) "miusuario"
["password"]=> string(10) “mipassword" ["database"]=> string(13) "mibasededatos" }
```