



IAI PHP Camp #2

Application Programming Interface API

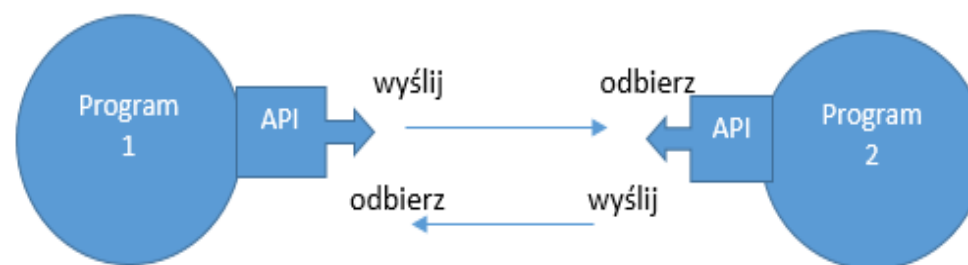
Cezary Dziekoński
Program Developer, IAI S.A.





Co to takiego?

zestaw reguł opisujących w jaki sposób dwa programy się ze sobą komunikują.





- Google Maps <http://www.google.com/apis/maps>
- FLICKR API <http://www.flickr.com/services/api>
- YOUTUBE <http://www.youtube.com/dev>
- EBAY <http://developer.ebay.com>
- AMAZON E-COMMERCE API <http://aws.amazon.com>
- WINDOWS LIVE MESSENGER API <http://dev.live.com>
- IdoSell Shop <http://www.idosell.com/pl/shop/api/>

Powody stosowania:

- ✓ Udostępnianie danych innemu, zewnętrznemu systemowi
- ✓ Udostępnianie danych modułowi w ramach tego samego systemu
- ✓ Asynchroniczne dostarczanie danych do serwisu internetowego
- ✓ Spójność danych - dane (np. klientów - imię, nazwisko, adres itp.) nie muszą być powielane w wielu systemach, dzięki czemu niwelowane jest ryzyko wystąpienia błędów (literówki, nieaktualne informacje itd.) w niektórych z nich.
- ✓ Wygoda użycia - dane wprowadzone bądź zmodyfikowane w jednym systemie są automatycznie aktualizowane w innych.



Ćwiczenie 1



Proszę napisać prosty skrypt :

1. Zadeklarować tablicę „products” zawierającą 5 tablic asocjacyjnych z polami(kluczami) id, nazwa, cena i wypełnić dowolnymi danymi.

2. Skrypt ma obsługiwać cztery parametry przekazane w GET:

product - wartość: numer od 1 do 10

action - jedna z wartości : checkProduct , addProduct, removeProduct

name – nazwa towaru

price – cena towaru

3. w przypadku akcji „checkProduct” pobrać towar na podstawie nazwy i wyświetlić jego dane

4. W przypadku akcji „addProduct” wstawić produkt z przekazaną id, ceną i nazwą i wyświetlić tablicę wynikową

5. W przypadku akcji „removeProduct” usunąć towar na podstawie przekazanego id i wyświetlić tablicę wynikową



<?php

```
$products = array(  
    array('id' => 3, 'nazwa' => 'towar1', 'cena' => 9.99),  
    array('id' => 4, 'nazwa' => 'towar2', 'cena' => 12.50),  
    array('id' => 6, 'nazwa' => 'towar3', 'cena' => 23.23),  
    array('id' => 7, 'nazwa' => 'towar4', 'cena' => 155.10),  
    array('id' => 12, 'nazwa' => 'towar5', 'cena' => 53.21)  
);  
  
$action = $_GET['action'];  
$id = $_GET['id'];  
$name = $_GET['name'];  
$price = $_GET['price'];  
  
if ($action === 'checkProduct') {  
    foreach ($products as $product) {  
        if ($product['nazwa'] === $name) {  
            break;  
        }  
    }  
    $result = $product;  
} elseif ($action === 'addProduct') {  
    $products[] = array('id' => $id, 'nazwa' => $name, 'cena' => $price);  
    $result = $products;  
} elseif ($action === 'removeProduct') {  
    foreach ($products as $index => $product) {  
        if ($product['id'] == $id) {  
            unset($products[$index]);  
            break;  
        }  
    }  
    $result = $products;  
} else {  
    $result = $products;  
}  
  
print_r($result);
```



API a PHP



Sposób przesyłania danych z pominięciem stron internetowych.



Wysyłanie żądań HTTP – cURL



curl – proste narzędzie konsolowe służące do wysłania żądań pod adresy URL.

Konsola: curl <http://www.google.com/>

W samym rdzeniu PHP jest rozszerzenie cURLa.

Przykład użycia:

```
$ch = curl_init('http://www.google.com');  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
$result = curl_exec($ch);
```




<?php

\$ch = curl_init('http://localhost/iaicamp/bramka.php?params=123');

curl_setopt(\$ch, CURLOPT_RETURNTRANSFER, true);

\$result = curl_exec(\$ch);

print_r(\$result);



Ćwiczenie 2



Proszę napisać skrypt php, który za pomocą cURLa wywoła program napisany w ćwiczeniu 1, zwrotek przypisze do zmiennej a następnie wyświetli zawartość zmiennej na ekranie.





JSON – JavaScript Object Notation – lekki i bardzo prosty format danych służący do reprezentowania zagnieżdżonych danych strukturalnych. Większość nowoczesnych języków programowania ma specjalne funkcje przeznaczone z tym formatem.

Przykład:

```
$zadania= array(`zakupy` =>array(`boczek`, `jajka`, `cebula`));
```

W formacie json:

```
{`zakupy` : { `boczek`, `jajka`, `cebula` }}
```

Funkcje do obsługi w PHP:

```
php => json = json_encode();
```

```
json => php = json_decode();
```



Kiedy używać:

Format JSON wyraźnie zaznacza, jaka była oryginalna struktura danych i przekazuje znajdujące się w niej wartości, ale nie informuje o tym jakie typy danych były używane. Formatu JSON warto używać wtedy, gdy format danych nie jest najistotniejszy i musi być łatwy do rozszyfrowania. Jest to idealny sposób przekazywania danych do Java Scriptu, gdzie w tym języku format JSON ma doskonałą i szybką obsługę.



Ćwiczenie 3



1. Program stworzony w ćwiczeniu 1 zmodyfikować tak, aby zwracał dane w formacie json.
2. Program stworzony w ćwiczeniu 2 zmodyfikować tak, aby pobrał dane z programu z punktu 1, odkodował je i wyświetlił w przejrzysty sposób wykorzystując tablicę php



<?php

```
$products = array(
    array('id' => 3, 'nazwa' => 'towar1', 'cena' => 9.99),
    array('id' => 4, 'nazwa' => 'towar2', 'cena' => 12.50),
    array('id' => 6, 'nazwa' => 'towar3', 'cena' => 23.23),
    array('id' => 7, 'nazwa' => 'towar4', 'cena' => 155.10),
    array('id' => 12, 'nazwa' => 'towar5', 'cena' => 53.21)
);

$action = $_GET['action'];
$id = $_GET['id'];
$name = $_GET['name'];
$price = $_GET['price'];

if ($action === 'checkProduct') {
    foreach ($products as $product) {
        if ($product['nazwa'] === $name) {
            break;
        }
    }
    $result = $product;
} elseif ($action === 'addProduct') {
    $products[] = array('id' => $id, 'nazwa' => $name, 'cena' =>
$price);
    $result = $products;
} elseif ($action === 'removeProduct') {
    foreach ($products as $index => $product) {
        if ($product['id'] == $id) {
            unset($products[$index]);
            break;
        }
    }
    $result = $products;
} else {
    $result = $products;
}

echo(json_decode($result));
```





<?php

```
$ch = curl_init('http://localhost/iaicamp/bramka_json.php');  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

```
$result = curl_exec($ch);
```

```
$response =json_decode($result, true);  
print_r($response);
```





XML – eXtensible Markup Language – kolejna standardowa technika prezentacji danych w formacie czytelnym dla maszyny.

- Mniej zwężły niż json
- Zawiera informacje o typach danych
- Zestaw znaczników i atrybutów może być inny dla każdego systemu
- Doskonale nadaje się do przesłania ważnych danych między dwoma systemami bez żadnego nadzoru

Przykład

```
<?xml version="1.0" ?>  
<list>  
  <item>jajka</item>  
  <item>chleb</item>  
  <item>cebula</item>  
</list>
```



```
<?php
```

```
header("Content-type: text/xml");
```

```
$list = array('jajka', 'boczek', 'cebula');
```

```
$simplexml = new SimpleXmlElement('<?xml version="1.0"?><list />');
```

```
foreach ($list as $item) {
```

```
    $simplexml->addChild('item', $item);
```

```
}
```

```
echo $simplexml->asXML();
```





```
<?php
```

```
$ch = curl_init('http://localhost/iaicamp/xml.php');  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
```

```
$result = curl_exec($ch);  
$list = simplexml_load_string($result);
```

```
echo "<table>\n";  
foreach ($list as $item) {  
    echo "<tr>\n";  
    echo "<td>". $item . "</td>";  
    echo "</tr>\n";  
}  
echo "</table>\n";
```



Ćwiczenie 4



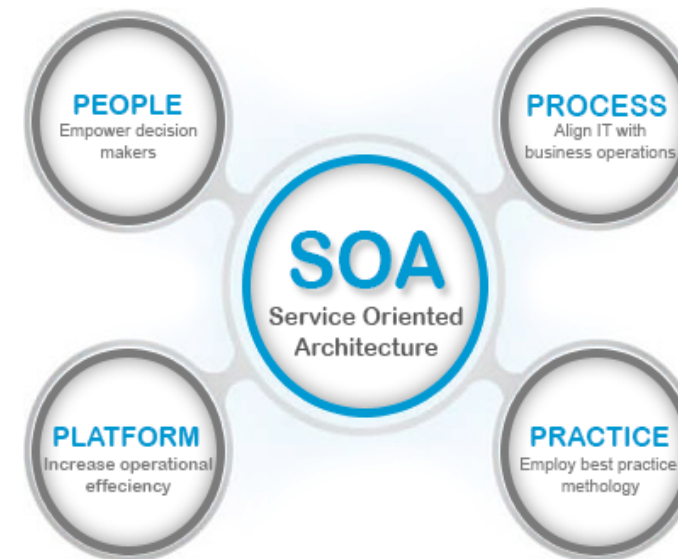
1. Program stworzony w ćwiczeniu 1 zmodyfikować tak, aby zwracał w formacie xml nazwy towarów
2. Program stworzony w ćwiczeniu 2 zmodyfikować tak, aby pobrał dane z programu z punktu 1, stworzył tablicę php z nazwami i wyświetlił je.



Usługi sieciowe



Bardziej rozległe i nowoczesne systemy projektowane są w taki sposób, że posiadają one swoją warstwę usługową, czyli funkcje systemu udostępniane są na zewnątrz systemu po to by mogły z nich korzystać inne programy, serwisy czy aplikacje działające na urządzeniach przenośnych.



Usługa RESTful



REST – Representational State Transfer – wygodna i prosta technika pozwalająca udostępnić operacje na danych według tzw. CRUD:

Create (put) - tworzenie

Read (get) pobranie

Update (put) – modyfikowanie

Delete (delete) – usuwanie

Operacje odbywają się na zasobach np. użytkownicy, produkty, zamówienia, kategorie. Dane które wysyłamy na dany zasób nazywana jest reprezentacja zasobu

Usługa RESTful c.d.



W RESTful działanie opiera się na strukturze adresu URL.

Wywołując:

<http://localhost/iaicamp/productslist/> - kolekcja (katalogi)

<http://localhost/iaicamp/productslist/product1> - konkretny zasób





IAI S.A. THE BEST
ONLINE
SELLING SOLUTIONS

Usługa RESTful - GET



```
<?php
```

```
$ch =curl_init('http://localhost/iaicamp/productslist');  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
$response = curl_exec($ch);  
$productsList = json_decode($response, 1);  
print_r($productsList);die;
```

Usługa RESTful - POST



```
<?php
$item = array('nazwa' => 'produkt13', 'id' => 321, 'cena' =>
12.22);

$data = json_encode($item);
$ch = curl_init('http://localhost/iaicamp/productslist');

curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);

$response = curl_exec($ch);
$productsList = json_decode($response, 1);
```





IAI S.A. THE BEST
ONLINE
SELLING SOLUTIONS

Usługa RESTful - PUT

```
<?php
$data = array(); // uzupełnić
$data = json_encode($boczek);
$ch =curl_init('http://localhost/iaicamp/productslist/produkt12');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "PUT");
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
$response = curl_exec($ch);
$productsList = json_decode($response, 1);
```





IAI S.A. THE BEST
ONLINE
SELLING SOLUTIONS

Usługa RESTful - DELETE

```
<?php
$ch = curl_init('http://localhost/iaicamp/productslist/produkt12');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_CUSTOMREQUEST, "DELETE");
$response = curl_exec($ch);
$productsList = json_decode($response, 1);
```



Ćwiczenie 5



1. Stworzyć API fullRest aplikacji która w zależności od wywołania wykona odpowiednią operację na zbiorze towarów (może być tablica z zadania 1) i zwróci wynik do skryptu wywołującego lub zapisze tablicę wynikową do pliku (file_put_contents()). Szkielet Api znajdź na kolejnym slajdzie.

2. Stworzyć cztery programy łączące się z API fullRest:

1. Stworzeniem nowego towaru (put)
2. Pobranem wszystkich towarów (get)
3. Modyfikacją nazwy i ceny jednego z towarów (post)
4. Usunięciem jednego z towarów (delete)





IAI S.A. THE BEST
ONLINE
SELLING SOLUTIONS



```
<?php
$products = array() ; // tablica zadeklarowana w pierwszym zadaniu

$path = $_SERVER['PATH_INFO'];
$pathArray = explode('/', $path);

$method = $_SERVER['REQUEST_METHOD'];

switch ($method) {
    case 'GET':
        // zwrotka na podstawie tablicy $pathArray
        break;
    case 'PUT':
        $params = json_decode(file_get_contents('php://input'), 1); // przekazane dane postem
        break;
    case 'POST':
        $params = json_decode(file_get_contents('php://input'), 1); // przekazane dane postem
        // zaktualizować dane
        break;
    case 'DELETE':
        // usunąć konkretny zasób na podstawie $pathArray
    default:
}

echo json_encode($products);
```

Rodzaje usług Sieciowych - SOAP



Soap SOAP (Simple Object Access Protocol) to protokół wymiany informacji pomiędzy różnymi hostami i aplikacjami sieciowymi. Jest niezależny od platformy i języka, gdyż wszelka wymiana danych odbywa się za pomocą powszechnie używanego języka XML.

SOAP jest standardem komunikacji pomiędzy potencjalnie nie związanymi ze sobą usługami sieciowymi. Serwer udostępniający usługę sieciową (Web Service) może np. działać pod kontrolą ASP.NET, natomiast aplikacja kliencka może być napisana w PHP, bądź w każdym innym dowolnym języku. Cała wymiana danych odbywa się bowiem poprzez niezależny od platformy język XML.

Jak działa SOAP? Otóż wyobraźmy sobie serwer, który może udostępniać daną usługę, np. podającą aktualną datę na serwerze. Serwer zatem posiada u siebie jakąś funkcję, która takową datę pobiera i może ją udostępnić na zewnątrz. Mając informację o tym jaki jest adres serwera i jak konkretnie taka funkcja się nazywa, aplikacja kliencka może wysłać do serwera żądanie z prośbą o wykonanie tej funkcji na serwerze i zwrócenia wygenerowanych przez nią danych do klienta.



```
<?php
class ServiceFunctions
{
    public function getInitials($first_name, $last_name)
    {
        $name = "";
        $name .= strtoupper(substr($first_name, 0, 1));
        $name .= ' ' . strtoupper(substr($last_name, 0, 1));
        return $name;
    }
}
```

```
$options = array('uri' => 'http://localhost/');
$server = new SoapServer(NULL, $options);
$server->setClass('ServiceFunctions');
$server->handle();
```





```
<?php
```

```
$options = array(  
'uri' => 'http://localhost',  
'location' => 'http://localhost/iaicamp/soap_server.php',  
'trace' => 1);
```

```
$client = new SoapClient(NULL, $options);  
echo $client->getInitials('Cezary', 'Dziekonski');
```



Ćwiczenie 6



Program stworzony w ćwiczeniu 1 zmodyfikować tak aby każda z akcji stała się funkcją obsługiwaną przez soap.





WSDL – Web Service Description Language – plik zawierający opis lokalizacji danej usługi, używanych przez nią typów danych oraz metod, parametrów. Używany jest przy wywołaniach SOAP.

WSDL: <http://trening6.iai-shop.com/api/?gate=checkserverload/checkServerLoad/39/soap/wSDL&lang=pol>

Dokumentacja: <https://www.iai-shop.com/pl/shop/api/?action=method&function=checkserverload&method=checkServerLoad>
[soap_wSDL_client.php]





Dziękuję za uwagę.

Prezentujący
Cezary Dziekoński
Program Developer, IAI S.A.



Zapraszamy na:
www.idosell.com