

Servicii <Web />

26LAICII <MGP />

Crearea de aplicatii Web
orientate spre servicii via REST

detalii in **[TX, 269-298] [SW, 61-244]**



“Concizia este sora talentului.”

Anton Cehov
АНТОН СЕХОВ



Exista o modalitate de invocare a serviciilor Web
fara a recurge la SOAP?



rest: representational state transfer

Stil arhitectural de dezvoltare a aplicațiilor Web
cu focalizare asupra reprezentării datelor

teza de doctorat (Roy Fielding, 2000)



rest
162f

Viziune complementara de implementare si
utilizare a serviciilor Web – fara SOAP

focalizare asupra reprezentarii datelor



rest
162f

Rezultatul unei procesari conduce la obtinerea
unei **reprezentari** a unei resurse



rest

162f

Rezultatul unei procesari conduce la obtinerea unei **reprezentari** a unei resurse

resursa Web = *blog*-ul unei persoane,
fotografia unui utilizator, flux de stiri, program etc.



rest
162f

Rezultatul unei procesari conduce la obtinerea
unei **reprezentari** a unei resurse

reprezentare = (X)HTML, JPEG, PNG, Atom, RSS, SVG,...



rest

162f

Rezultatul unei procesari conduce la obtinerea unei **reprezentari** a unei resurse

reprezentare = (X)HTML, JPEG, PNG, Atom, RSS, SVG,...

desemnată de **tipuri MIME**
text/html, image/png, application/json etc.

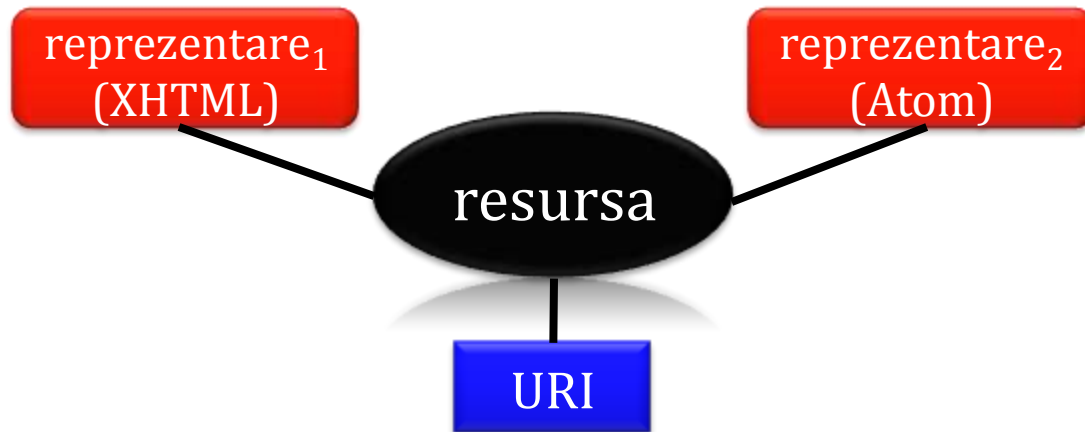


rest

162f

Rezultatul unei procesari conduce la obtinerea unei **reprezentari** a unei resurse

reprezentarile aceleasi resurse
– desemnate de un URI unic – pot fi multiple



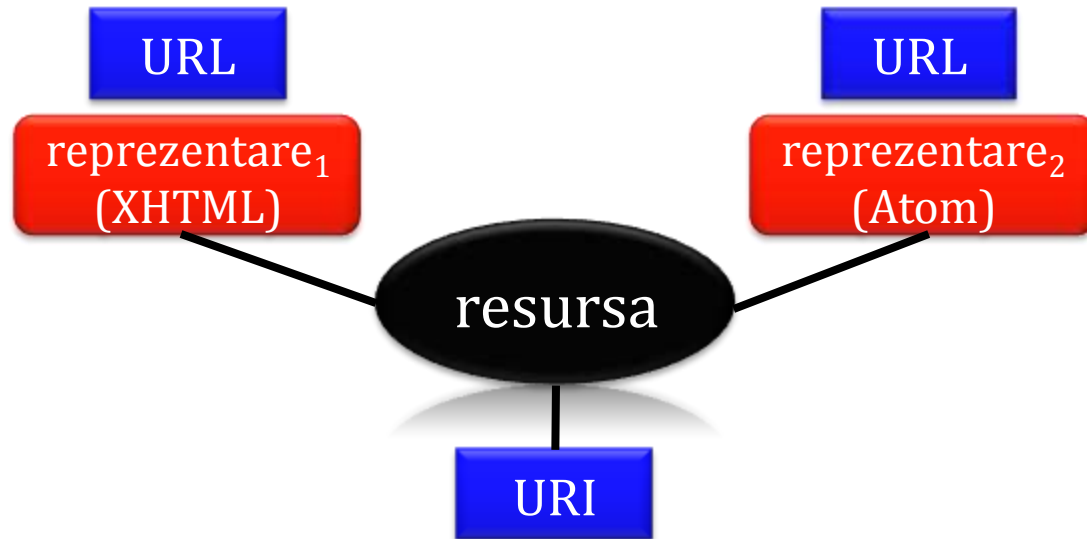


rest

162f

Rezultatul unei procesari conduce la obtinerea unei **reprezentari** a unei resurse

fiecare reprezentare a unei resurse are asociat un URL

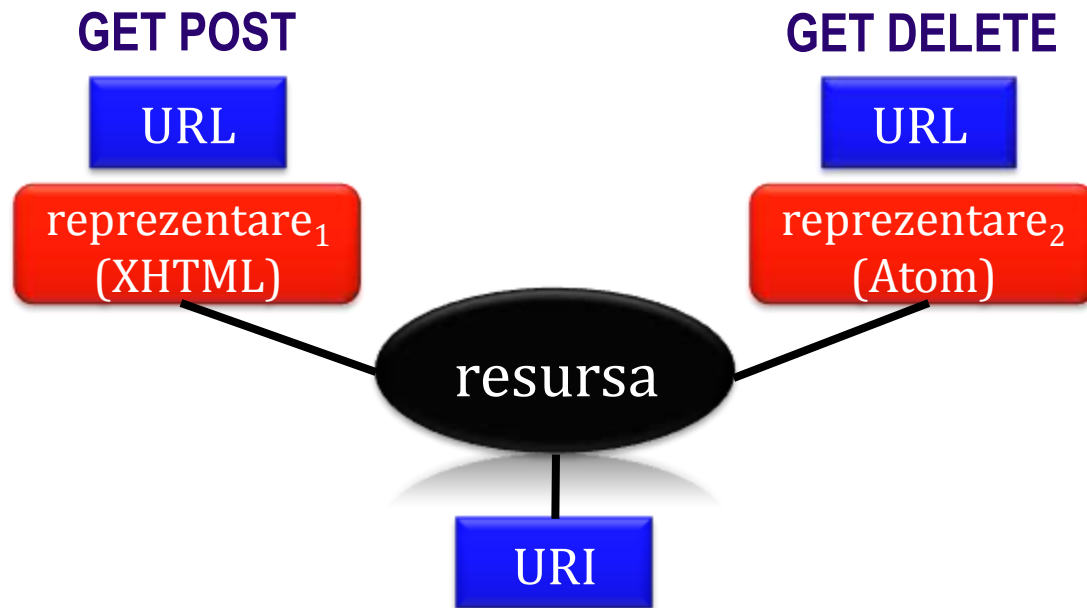




rest

162f

Clientii (*e.g.*, navigatoare Web, roboti, *player*-e etc.) interactioneaza cu reprezentarile resurselor via **verbe** “acceseaza”: **GET**, “modifica”: **POST**, “sterge”: **DELETE**,...

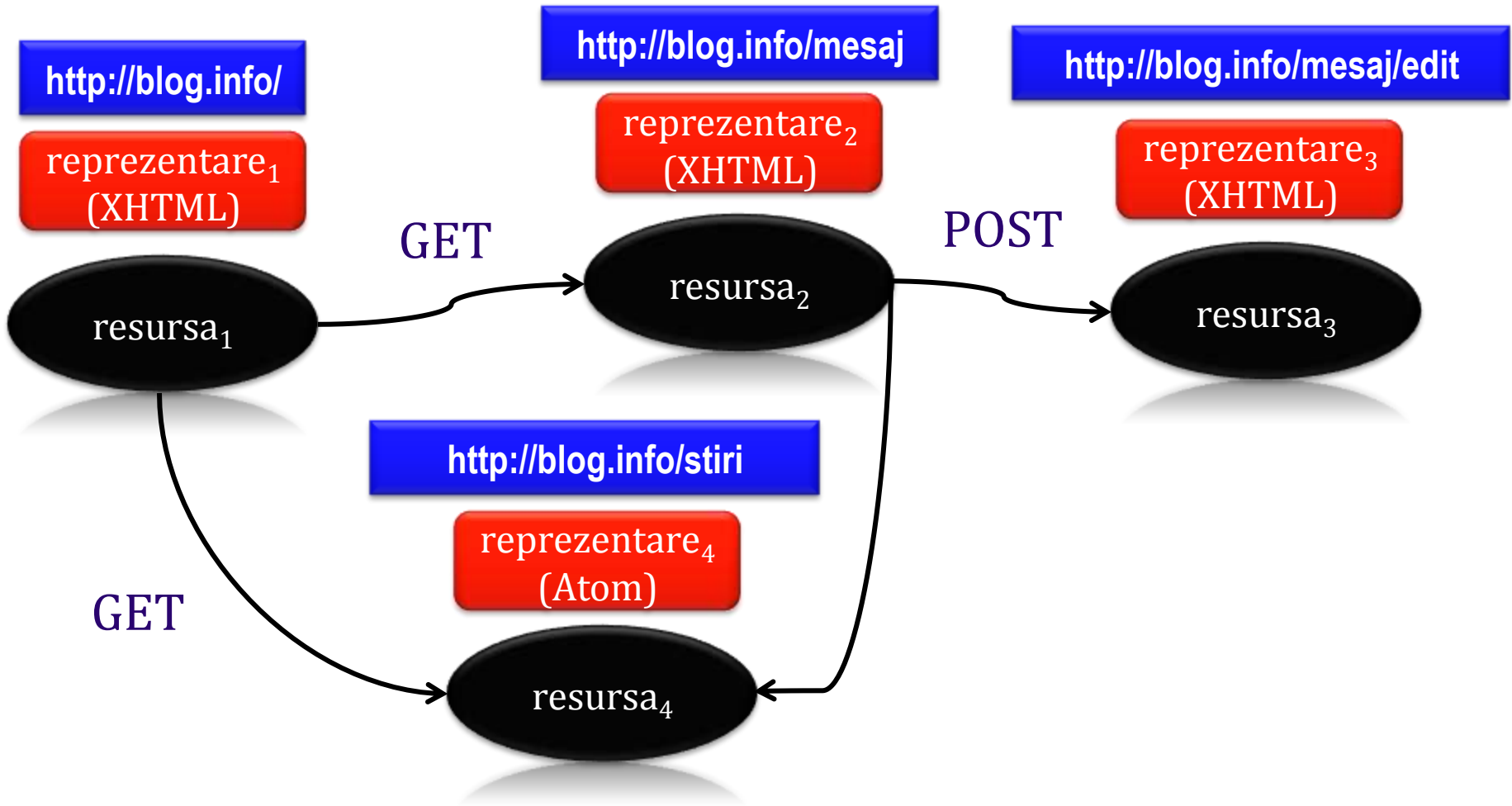




rest

162f

Orice accesare a unei reprezentari plaseaza aplicatia – ori clientul Web – intr-o **stare** ce va fi schimbata in urma unui **transfer** de date (accesarea altei reprezentari)





rest
162f

Transferul se realizeaza prin protocolul **HTTP**

Reprezentarea este modelata in **XML**
(sau alt format) si indicata prin tipuri **MIME**

Adresabilitatea se rezolva via **URI**



rest
162f

Aplicatiile care invoca functionalitati (servicii)
consuma reprezentari de resurse – in stilul *pull*



rest

162f

Fiecare cerere este considerata independenta, fara a se lua in considerare contextul – *stateless*



rest: exemplu
rest: exemplu

Implementarea unui magazin *on-line*
oferind dulciuri



rest: exemplu

Implementarea unui magazin *on-line*
oferind dulciuri

stilul “clasic” **SOAP** – conform RPC

operatii privind produsele:

furnizeazaSortim(), adaugaSortim(), listeazaSortim(), cautaSortim()

operatii ce vizeaza managementul utilizatorilor:

furnizeazaUtiliz(), adaugaUtiliz(), eliminaUtiliz(), cautaUtiliz(),...



rest: exemplu

REST: EXEMPLU

Implementarea unui magazin *on-line*
oferind dulciuri

stilul “nou” **REST**

tipuri de resurse (**Sortim** & **Utiliz**), identificate unic de URI
<http://www.dulciuri.biz/sortim/jeleu>



rest: exemplu

Implementarea unui magazin *on-line*
oferind dulciuri

stilul “nou” **REST**

tipuri de resurse (**Sortim** & **Utiliz**), identificate unic de URI

<http://www.dulciuri.biz/sortim/jeleu/cu-gust-de-web>

URI intuitiv – “*user/SEO friendly*”



rest: exemplu real

Serviciile puse la dispozitie de **Delicious**

managementul adreselor Web favorite (*bookmark-uri*),
cu posibilitatea atasarii de termeni de continut (*tag-uri*)
si comentarii

social bookmarking

<https://delicious.com/developers>



rest: exemplu real

Serviciile puse la dispozitie de **Delicious**

- functionalitate de baza:
 - listarea tuturor *bookmark*-urilor
(eventual, filtrate dupa diverse criterii)
- managementul *bookmark*-urilor:
 - adaugare, editare, stergere

rest: exemplu real

Resursa	URL	Metoda	Reprezentare
<i>Bookmark</i>	<code>/bookmarks/{md5}</code>	GET	<code>application/bookmark+xml</code>
<i>Bookmark</i>	<code>/bookmarks/{md5}</code>	PUT	<code>application/bookmark+xml</code>
<i>Bookmark</i>	<code>/bookmarks/{md5}</code>	DELETE	
Lista de <i>bookmark</i> -uri	<code>/bookmarks</code>	GET	<code>application/atom+xml</code>
Lista de utilizatori	<code>/users</code>	GET	<code>application/atom+xml</code>
Lista de <i>tag</i> -uri (termeni de continut)	<code>/tags</code>	GET	<code>application/atom+xml</code>
Pagina principala	<code>/</code>	GET	<code>application/delicious+xml</code>

GET /bookmarks

200 OK

Content-type: application/atom+xml...

raspuns XML (Atom)
oferit de serviciul Delicious

<?xml version="1.0"?>

<feed xmlns="http://www.w3.org/2005/Atom">

<title>Bookmarks</title>

<entry>

<title>O resursa interesanta</title>

<link

href="/bookmarks/a211528fb5108cddaa4b0d3aecbdbcf"/>

<summary>

http://undeva.info/o-resursa-interesanta

</summary>

</entry>

<!-- eventual, alte elemente <entry>... -->

</feed>

digest MD5

obtinerea
bookmark-urilor

GET /bookmarks/a211528fb5108cddaa4b0d3aecdbdcf
200 OK

Content-type: application/bookmark+xml

```
<bookmark>
  <title>O resursa interesanta</title>
  <url>http://undeva.info/o-resursa-interesanta</url>
  <user href="/users/tux">tux</user>
  <tags>
    <tag href="/tags/interesting">interesting</tag>
    <tag href="/tags/penguin">penguin</tag>
  </tags>
</bookmark>
```

preluarea unui *bookmark*:
răspunsul XML dat de serviciul **Delicious**

POST /bookmarks

Content-type: application/bookmark+xml

...

201 Created

Location: /bookmarks/a211528fb5108cddaa4b0d3aeccdbdcf

crearea unui
bookmark

PUT /bookmarks/a211528fb5108cddaa4b0d3aeccdbdcf

Content-type: application/bookmark+xml

...

200 OK

actualizarea
unui *bookmark*

```
<posts user="busaco" tag="">
  <post href="http://interact.webstandards.org/"
    hash="a4dff89d5c32649fdf943d392ff75b73"
    description="WaSP InterAct Curriculum"
    tag="standards tutorial web css accessibility development"
    time="2010-03-11T07:29:21Z" />
  <post href="http://www.opera.com/company/education/curriculum/"
    hash="f060f790348eb6027795bafc2a7b739e"
    description="Opera: Web Standards Curriculum"
    tag="css web html standards reference development javascript"
    time="2010-03-11T07:27:30Z" />
  <post href="http://code.google.com/p/linked-data-api/wiki/Specification"
    hash="d3f1a47793164457566e666df8027718"
    description="Specification - Linked Data API Specification"
    tag="linkeddata api rdf semantic web lod programming sparql"
    time="2010-03-01T13:28:14Z" />
</posts>
```

fragment al documentului XML
specificand *bookmark*-urile unui utilizator

```
<posts user="busaco" tag="">
  <post href="http://interact.webstandards.org/"
    hash="a4dff89d5c32649fdf943d392ff75b73"
    description="WaSP InterAct Curriculum"
    tag="standards tutorial web css accessibility development"
    time="2010-03-11T07:29:21Z" />
  <post href="http://www.opera.com/company/education/curriculum/"
    hash="f060f790348eb6027795bafc2a7b739e"
    description="Opera: Web Standards Curriculum"
    tag="css web html standards reference development javascript"
    time="2010-03-11T07:27:30Z" />
  <post href="http://code.google.com/p/linked-data-api/wiki/Specification"
    hash="d3f1a47793164457566e666df8027718"
    description="Specification - Linked Data API Specification"
    tag="linkeddata api rdf semantic web lod programming sparql"
    time="2010-03-01T13:28:14Z" />
</posts>
```

format xsd:date (XML Schema)

fragment al documentului XML
specificand *bookmark*-urile unui utilizator



rest

162f

Resursele sunt numite folosind URI-uri (URL-uri)

Reprezentarile sunt interconectate prin URL-uri

Pot exista intermediari (*proxy, cache, porti*)
intre clienti si resurse ► performanta, securitate,...



rest

162f

Resursele sunt numite folosind URI-uri (URL-uri)

Reprezentarile sunt interconectate prin URL-uri

Pot exista intermediari (*proxy, cache, porti*)
intre clienti si resurse ► performanta, securitate,...

Transferul de date poate fi si asincron – in stilul Ajax

vezi cursul viitor



rest

162f

O resursa poate avea asociate reprezentari (XML, JSON, eventual in alte formate de date) ce pot fi accesate/alterate via operatii HTTP

operatii **CRUD** – *Create, Retrieve, Update, Delete*



rest: metodologie

Divizarea in resurse a setului de date
ale problemei

clase tipice de resurse:

Utilizator

Document – alternative: **Fotografie, Produs, Software, ...**

Metadata – *e.g.*, **Comentariu, Format, Locatie, Platforma** etc.



rest: metodologie

Numirea prin URI a fiecărei resurse

exemplificari:

<http://aplicatie.info/Utilizator/tux>

<http://aplicatie.info/Document/pinguini-cu-mere-albastre>



rest: metodologie

Numirea prin URI a fiecărei resurse

cazuri concrete:

insemnarile etichetate cu **fii** disponibile pe un *blog*

<http://fiistudent.wordpress.com/tag/fii/>

accesarea datelor despre o productie cinematografica

<http://www.imdb.com/title/tt0401383/>

obtinerea listei utilizatorilor ce urmaresc o persoana

<http://twitter.com/followers>



rest: metodologie

Numirea prin URI a fiecărei resurse

cazuri concrete:

acces la prezentările Slideshare ale utilizatorului **busaco**

<http://www.slideshare.net/busaco/presentations>

bookmark-urile cu *tag*-ul **web** ale utilizatorului **busaco**

<http://delicious.com/busaco/web>

efectuarea acțiunii de editare a unui document XWiki

<http://www.info.uaic.ro/bin/edit/Students/Timetable>



rest: metodologie

Proiectarea reprezentarii(lor) *acceptate*
ce pot fi trimise de aplicatia client
si reprezentarii(lor) *intoarse* spre client

de considerat formatele standard – *e.g.*, HTML, Atom



rest: metodologie

Integrarea resurselor
via legaturi hipertext + formulare



rest: metodologie

Crearea de studii de caz

specificarea conditiilor de eroare si/sau de exceptie



Cum pot fi accesate reprezentari de resurse Web
prin REST?



rest: implementare

Biblioteci/API-uri implementând HTTP

libcURL (C, cu portări pentru Perl, PHP, Ruby,...)

libwww (C, Perl)

httplib (Python)

Net:HTTP (Ruby)

RestSharp (C#)

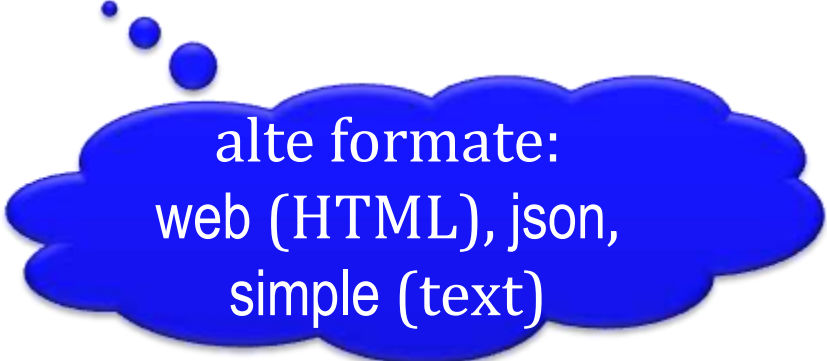
etc.



rest: implementare

Studiu de caz: invocarea unui serviciu Web
de prescurtare de URL-uri – <http://is.gd/>

un nou URL prescurtat va fi creat folosind adresa
<http://is.gd/create.php?format=xml&url=adresaWeb>



alte formate:
web (HTML), json,
simple (text)

Cererea HTTP ce invoca serviciul Web prin REST:

```
GET /create.php?format=xml&url=profs.info.uaic.ro/~busaco HTTP/1.1  
Host: is.gd
```

Raspunsul obtinut, transmis de serverul Web:

```
HTTP/1.1 200 OK  
Server: nginx  
Date: Tue, 15 May 2012 07:48:18 GMT  
Content-Type: text/xml;charset=UTF-8
```



reprezentare POX
(*Plain Old XML*)

```
<?xml version="1.0" encoding="UTF-8" ?> . . .  
<output>  
  <shorturl>http://is.gd/DBmG2L</shorturl>  
</output>
```

```
// initializam cURL
$c = curl_init ();
// stabilim URL-ul serviciului Web invocat
curl_setopt ($c, CURLOPT_URL,
    'http://is.gd/create.php?format=xml&url=profs.info.uaic.ro/~busaco');
// rezultatul cererii va fi disponibil ca sir de caractere
curl_setopt ($c, CURLOPT_RETURNTRANSFER, 1);
// preluam resursa oferita de server (aici, un document XML)
$res = curl_exec ($c);
// inchidem conexiunea cURL
curl_close ($c);
// procesam rezultatul via DOM
$doc = new DOMDocument ();
$doc->loadXML ($res);
// preluam continutul elementului <shorturl>
$urls = $doc->getElementsByTagName ('shorturl');
foreach ($urls as $url) {
    echo '<p>Adresa prescurtata este: ' . $url->nodeValue . '</p>';
}
```



apelarea serviciului
Web REST via PHP



rest: implementare

Twitter / Interactions

http://localhost:8888/rest.php

Invocam serviciul Web de la <http://is.gd/create.php?format=xml&url=profs.info.uaic.ro/~busaco>

Raspunsul oferit de serviciu:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><output>  
<shorturl>http://is.gd/DBmG2L</shorturl></output>
```

Adresa prescurtata este: <http://is.gd/DBmG2L>



rest: implementare

Studiu de caz: accesarea celor mai recente mesaje disponibile pe Twitter

se recurge la URL-ul

http://twitter.com/statuses/public_timeline.json

pentru a obtine mesajele publice
in format JSON (*JavaScript Object Notation*)

detalii la <https://dev.twitter.com/docs/api>

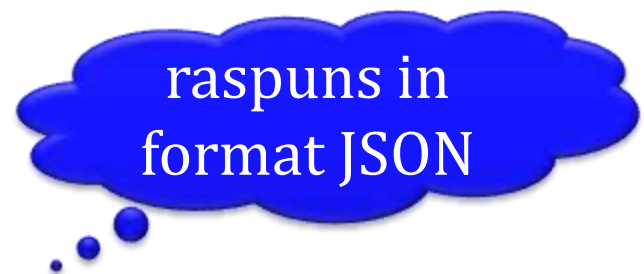
HTTP/1.1 200 OK

Date: Tue, 15 May 2012 07:03:27 GMT

Content-Type: **application/json**; charset=utf-8

Content-Length: 40860

```
[ {  
  ...  
  "created_at": "Tue May 15 07:03:23 +0000 2012",  
  "truncated": false,  
  "retweet_count": 0,  
  "retweeted": false,  
  "in_reply_to_user_id": null,  
  "user": {  
    "name": "Tuxy Pinguinescu",  
    "location": "South Pole",  
    "created_at": "Wed Oct 21 06:33:14 +0000 2010",  
    "lang": "en",  
    "followers_count": 133374,  
    ...  
  },  
  "in_reply_to_status_id": null  
}
```



raspuns in
format JSON



rest: implementare

Biblioteci/API-uri implementând HTTP

permit dezvoltarea de aplicații *desktop*, *mobile* etc.

suport pentru crearea de *mash-up*-uri la nivel de server

nu funcționează în navigatorul Web



rest: implementare

Biblioteci/API-uri implementând HTTP

permit dezvoltarea de aplicații *desktop*, *mobile* etc.

suport pentru crearea de *mash-up*-uri la nivel de server

nu funcționează în navigatorul Web

atenție la problemele de securitate ce pot apărea!



rest: implementare

Navigatoarele Web actuale

nu necesită o interfață de programare (API) specifică
disponibilitate pe orice platforma

suport pentru REST via obiectul **XMLHttpRequest** (Ajax)

dependența de implementarea la nivel de browser



rest: implementare

Exemplificare recurgand la biblioteca jQuery (JavaScript):

```
jQuery.ajax ({ // executa o cerere POST pentru invocarea serviciului Web
  type: "POST",
  contentType: "application/json; charset=utf-8",
  url: "http://undeva.info/ServiciuWeb/Resursa",
  data: "{...}", // datele de intrare trimise serviciului
  dataType: "json", // asteptam raspunsul in format JSON
  success: function(data) { // functie apelata la transfer cu succes
    $(' .rezultat').html(data); // preluam datele, convertindu-le in HTML
  }
});
```

soap vs. rest

209b 12' 162f

SOAP	REST
Actiuni arbitrare (verbe)	Actiuni fixe – HTTP: GET, POST,...
Structuri de date oricit de complexe – via XML Schema	Opereaza asupra reprezentarilor de resurse (<i>i.e.</i> , XML, HTML)
Descriere complexa a serviciului (pe baza WSDL)	Scalabil (facil de extins)
Suport pentru <i>XML messaging</i>	Bazat pe URI
Dezvoltare sofisticata: securitate, intermediari, specificatii WS-*, interoperabilitate,...	Uzual, mai usor de programat (+disponibilitatea API-urilor)
Specific mediului <i>enterprise</i>	Caracteristic aplicatiilor sociale (Web 2.0)



rest: dezvoltare

rest: dezvoltare

ASP.NET MVC (C#)

JAX-RS – *Java Architecture for RESTful web Services*

Jifty (Perl)

neon (C)

Restlet (Java)

RIP – *REST in Python*

Ruby on Rails (Ruby)

Recess, Symfony, Tonic, Zend Framework (PHP)

etc.



rest: dezvoltare

REST: dezvoltare

Servicii publice ce pot fi consumate via REST – exemple:
500px, Amazon, Basecamp, Blip.tv, eBay, Ericsson,
Facebook, GitHub, Google, Klout, LinkedIn, Muselius,
Netflix, OpenSocial, Scribd, SlideShare, TripIt, Yahoo! etc.



3986 API-uri specifice (20 mai 2012)

vezi si <http://tinyurl.com/2ssfc2>

disponibile pentru ActionScript, C++, C#, JavaScript,
Java, PHP, Python, Objective-C, Ruby,...

http://github.com/api/v2/json/repos/show/victorporof/Tilt

GET



follow redirects

no auth HTTP basic

+ add header

Send

testarea de API-uri Web cu hurl.it

request

response

[view full size](#) [permalink](#)

```
HTTP/1.1 200 OK
Server: nginx/1.0.13
Date: Sat, 19 May 2012 16:00:44 GMT
Content-Type: application/json; charset=utf-8
Connection: close
Status: 200 OK
X-RateLimit-Limit: 60
ETag: "5da500c7754d142b984a6059f23a2661"
X-Frame-Options: deny
X-RateLimit-Remaining: 59
X-Runtime: 273
Content-Length: 447
Cache-Control: private, max-age=0, must-revalidate
```

```
{
  "repository": {
    "name": "Tilt",
    "created_at": "2011/05/25 11:00:15 -0700",
    "size": 236,
    "has_wiki": true,
    "watchers": 366,
    "private": false,
    "language": "JavaScript",
    "fork": false,
    "url": "https://github.com/victorporof/Tilt",
```

accesarea
serviciului Web
oferit de GitHub

JSON



rest: dezvoltare

rest: dezvoltare

Pasi uzuali de urmat pentru implementarea unei aplicatii ce va invoca un serviciu Web pe baza unui API public:

(1) inregistrarea aplicatiei concepute
via situl entitatii furnizoare a serviciului

▶ cheie de acces – *API key, consumer key*



rest: dezvoltare

162f: q6z\01f916

Pasi uzuali de urmat pentru implementarea unei aplicatii ce va invoca un serviciu Web pe baza unui API public:

(2) pe baza acestei chei, aplicatia se va putea autentifica pentru a putea fi autorizata sa acceseze serviciul



rest: dezvoltare

rest: dezvoltare

Pasi uzuali de urmat pentru implementarea unei aplicatii ce va invoca un serviciu Web pe baza unui API public:

(2) pe baza acestei chei, aplicatia se va putea autentifica pentru a putea fi autorizata sa acceseze serviciul

pot fi impuse diverse politici de acces (*permissions*): doar consultare (*read*), posibilitatea editarii etc.



rest: dezvoltare

162f: q6z\01f916

Pasi uzuali de urmat pentru implementarea unei aplicatii ce va invoca un serviciu Web pe baza unui API public:

- (3) autentificarea si autorizarea aplicatiei
au loc cu acordul utilizatorului



rest: dezvoltare

rest: dezvoltare

Pasi uzuali de urmat pentru implementarea unei aplicatii ce va invoca un serviciu Web pe baza unui API public:

(3) autentificarea si autorizarea aplicatiei
au loc cu acordul utilizatorului

daca utilizatorul nu este autentificat, i se vor solicita informatiile de autentificare (*e.g.*, nume + parola), apoi va putea autoriza aplicatia sa aiba acces la date via serviciul Web furnizat



rest: dezvoltare

rest: dezvoltare

Pasi uzuali de urmat pentru implementarea unei aplicatii ce va invoca un serviciu Web pe baza unui API public:

- (4) aplicatia apeleaza metodele oferite de serviciu pentru preluarea/modificarea datelor de interes, conform politicilor de acces



rest: dezvoltare

162f: q6z\01f916

Pasi uzuali de urmat pentru implementarea unei aplicatii ce va invoca un serviciu Web pe baza unui API public:

(4) aplicatia apeleaza metodele oferite de serviciu

sesiunea curenta va fi stabilita si mentinuta pe baza unor **informatii de autentificare** (*auth tokens*)



rest: dezvoltare – oauth

rest: dezvoltare – oauth

Autorizarea securizata a unei aplicatii sa acceseze date private intr-un mod standardizat se poate realiza via **OAuth**

rest: dezvoltare – oauth

rest: dezvoltare – oauth

Autorizarea securizata a unei aplicatii sa acceseze date private intr-un mod standardizat se poate realiza via **OAuth**

protocol deschis

OAuth 1.0 – RFC 5849 (2010), OAuth 2.0 (in lucru)

<http://oauth.net/>

<http://hueniverse.com/oauth/>



rest: dezvoltare – oauth

rest: dezvoltare – oauth

Exemplificari concrete:

Evernote, Facebook, Google, Ohloh, OpenSocial,
Photobucket, TripIt, Twitter, Vimeo, Windows Live,
Yahoo!, Yammer,...



rest: dezvoltare – openid
162f: q6z\o1f916 – ob6niq

OpenID

mecanism descentralizat de autentificare a utilizatorului la nivel de Web pe baza paradigmei SSO – *Single Sign On*

via un ofertant (serviciu) de identitate (*identity provider*), utilizatorii pot sa-si gestioneze prezente *online* multiple

<http://openid.net/>



rest: dezvoltare – openid

OpenID

fiecare identitate e desemnata de un URL (stabilit de *identity provider*): [me.yahoo.com](#), [nume.wordpress.com](#), [google.com/profiles/nume](#), [launchpad.net/~nume](#) etc.

pentru a-si confirma identitatea,
utilizatorul va trebui sa se autentifice:
nume de cont + parola, *smart card*, date biometrice,...



rest: dezvoltare – openid
rest: dezvoltare – openid

OpenID

Biblioteci *open source* disponibile
pentru C#, Java, Perl, PHP, Python, Ruby,...

<http://openid.net/developers/libraries/>
<http://www.janrain.com/openid-enabled>



rezumat

LS301143

Accesarea serviciilor Web prin REST
consideratii privind autentificarea & autorizarea

