

Amazon webszolgáltatások

Egyetlen URL-lel webszolgáltatások széles tárházát vehetjük igénybe

Egyetemista koromban nem sok lehetőség volt szakkönyvek beszerzésére. Új példányokat a nagyon drága egyetemi könyvesboltban lehetett megvenni vagy a nagyon drága konkurenciánál, vagy használtakat más diákoktól, akik egy-egy félév végén mindig kiárúsítást csináltak. Könyvvásárlási szokásaimat tehát a lakókörnyezetem határozta meg, ráadásul mindig azt tanultam, ami elérhető volt. Ezek után talán nem meglepő, hogy amint lehetővé vált, beálltam az online könyvesboltok vásárlóinak sorába. Ilyen boltok a *Bookpool* és az *Amazon*, őket legalább 1995 nyara óta „támogatom” vásárlásaimmal. Valóra vált álom, mert a kitűnő árak mellett széles a választék, s a kényelem sem utolsó szempont. A gond csak az, hogy pénzt még mindig csak annyit költhettem így, mint a hagyományos könyvesboltokban. Szóval a könyvekre költhető anyagi keretem változatlan maradt, az ebből megvehető könyvek száma, valamint a választék az arányokat tekintve igen távol került egymástól.

A helyzet jobbra fordult, mikor az *Amazon* megnyitotta kapuit más könyvkereskedők előtt. Nem csak arra nyílt lehetőség, hogy összehasonlítsam az új könyvek árát saját asztalomnál ülve, de böngészhettem régi könyvek között és vásárolhattam is. Nem egészen egy dollárért (plusz postaköltségért) számtalan érdekes könyvből válogathattam, így lassan valamiféle könyvzabáló szörnyeteggé váltam. Könyvespolcomon gyűltek a könyvek, amelyek – reméltem – a kutatásaimhoz segítséget nyújtanak majd, de főleg azért vásároltam őket, mert volt rá lehetőségem. Most is, ha

valami jót hallok egy könyvről, első reakcióm az, hogy utánanézek az *Amazon*on – vagy még jobb, az *isbn.nu*-n, ahol össze lehet hasonlítani a különböző oldalakon található árakat.

Az *Amazon* az évek során hatalmas, temérdek könyv adatait tároló adatbázist épített fel. Biztos vagyok benne, hogy ez a könyv, vásárlói és eladói adatbázis továbbra is nagy segítségére van az *Amazon* döntéshozóinak. Néhány éve azonban az *Amazon* egy meglepő dolgot tett: belső adatbázisuk egy részét megnyitották más fejlesztők számára az *Amazon Web Services (AWS)* program keretében. Így, az *AWS*-szel a fejlesztők bármilyen feladatot végrehajthatnak, amit az *Amazon* oldalon is lehet, csak nem webböngészőt használnak a célra, hanem egy ügyféloldali programot. Az *AWS* a könyvkereskedőknek néhány más szolgáltatást is ajánl árkialakítással és beszerzés-ügyintézésel kapcsolatban.

2005 második felében az *Amazon* számos olyan új webszolgáltatást vezetett be, amelyek már csak részben köthetők az adásvételhez. Ugyanebben az időben az *eBay* bejelentette, hogy nem kér ellenszolgáltatást webszolgáltatásainak használatáért, így két, a világ legnagyobbjai között levő kereskedelmi adatbázis vált elérhetővé. Természetesen a *Google* is régóta kínál saját webszolgáltatásokat; és annak ellenére, hogy az adatok jelenleg a főindexre vannak korlátozva, azért kijelenthetjük, hogy remek adatforrás.

Ebben a hónapban felfedező útra indulunk a kereskedelmi webszolgáltatások között, és olyan megoldásokat keresünk, amelyekkel

a külső webszolgáltatások adatait saját alkalmazásaink részévé tehetjük. Az út során megismerjük a webszolgáltatások elérésének különböző módjait, a rendelkezésünkre álló lehetőségeket, és azt, hogyan hozhatunk létre meglevő webszolgáltatásokon alapuló új, érdekes alkalmazásokat.

Mit nevezünk webszolgáltatásnak?

A web első évtizedében annak célja főleg a felhasználókkal való interakció volt. Azaz a legtöbb *HTTP* ügyfél webböngésző volt, az általuk letöltött tartalom zöme pedig emberi fogyasztásra szánt *HTML*-formátumú szöveg. Aztán egyszer csak a fejlesztők felismerték a *HTTP*-ben rejlő lehetőségeket, nevezetesen, hogy az nem csak emberi olvasásra való dokumentumok továbbítására használható. Megkezdődött a *HTTP* programok közötti adatátvitelre való használata. A *HTTP*, mint átviteli protokoll és az *XML*, mint adatformátum kombinációja az *XML-RPC*-hez vezetett. Mivel mind az *XML*, mind a *HTTP* platformsemleges, nem volt szükség arra, hogy ugyanazon a nyelven írják meg az ügyfél- és a kiszolgálóoldali programokat, és arra sem, hogy ugyanazt az operációs rendszert használják. Ily módon az *XML-RPC* egy platformok közötti *RPC*-lehetőséget (*RPC: remote procedure call, távoli eljáráshívás*) biztosít, és sokkal kevesebb többletterheléssel jár, mint ugyanezeknek a problémáknak más hasonló megközelítései (mint például a *CORBA* közvetítő szoftver). Az *XML-RPC* egy jó, tiszta és könnyűsúlyú protokoll, de sok fejlesztő szerint nem elég kifinomult és vannak hiányosságai a hibakezelés

és az adattípusok terén is. A *SOAP* (a *Simple Object Access Protocol* rövidítése, magyarul egyszerű objektum-hozzáférési protokoll) számos újítást bevezetett, többek között azt, hogy elkülöníti egymástól az üzenet boríték és törzs részét.

Az *XML-RPC*-hez és a *SOAP*-hoz is szükséges egy kiszolgáló, amely megadott *URL*-eken figyel a függvényhívásokra. A kiszolgáló figyelhet mondjuk a */server/queries*, vagy valami hasonló *URL*-en. Az ügyfélnek jeleznie kell a kérésben, hogy milyen tagfüggvényt akar meghívni. *XML-RPC*-ben a *methodName* címkét használjuk és egy *HTTP POST* kéréssel elküldött *XML* borítékban adjuk át az összes paramétert és metaadatot. Egy másik technika, a *REST*, a függvényhívásokat magában az *URL*-ben azonosítja. A paramétereket szabványos *GET* kérésként adja át. A *REST* számos jó tulajdonsága közül a legfontosabb az egyszerűsége és könnyen használhatósága. Igen könnyű vele a hibakeresés is, mert az *URL*-eket akár egy webböngészőbe is beírhatjuk. Ennek ellenére még mindig sokan használják a *SOAP*-ot és az *XML-RPC*-t, főleg összetett adatszerkezetek kezelésére.

Az webszolgáltatások képezik az alapját azoknak a technológiáknak, melyeket az utóbbi időben *SOA*-ként, azaz szolgáltatás-orientált architektúráként (*Service-Oriented Architecture*) emlegetnek. A webszolgáltatások a web összes erősségét kihasználják, úgy mint: platform- és programozási nyelv függetlenség, valamint az a képesség, hogy új változat terjesztése nélkül frissíthetők és változtathatók a szolgáltatások.

A *SOA* lehetővé teszi, hogy új szolgáltatásokat hozzunk létre, vagy akár a már meglévőknél vezetjük be új változatait akár egy meglévő megvalósítás helyettesítésével, akár egy, a régi-vel párhuzamos új változat létrehozásával. A webszolgáltatások felhasználói hasznot húzhatnak a megnövekedett sebességből és hatékonyságból, vagy a teljesen új programozói felületekből anélkül, hogy az inkompatibilitás vagy a telepítés gondot okozna nekik. Ezenkívül amíg a fejlesztők a szolgáltatások közzétett specifikációit követik, addig bármilyen nyelvet és platformot használhatnak, és bármit létrehozhatnak

a párbeszédés asztal alkalmazástól kezdve az automatizált kötegelt feladatokig, amelyek több gigabájt adaton ráják át magukat.

Az Amazon webszolgáltatásai

Az *Amazon* az elsők között kezdett webszolgáltatásokat nyújtani. Az *AWS* egy különböző programozói felületekből álló programcsomag, melynek bizonyos részei az *Amazon* adatbázisához kapcsolódnak, mások, mint a *Mechanical Turk*, vagy az *Amazon's Simple Queue Service* általánosabb célokra valók. A legnépszerűbb szolgáltatás az *E-Commerce Service* (*ECS*, elektronikus kereskedelmi szolgáltatás), amely lehetővé teszi, hogy a termékek adatait kikeressük az *Amazon* különböző depóiból, részletes információkhoz jussunk egy-egy adott termékről vagy terjesztőről, és alapvető e-kereskedelmi műveleteket hajthassunk végre, például, hogy létrehozhassunk kosarat és módosíthassuk annak tartalmát.

Az *ECS*-nek két alapvető működési módja van, a keresés (*search*) és a kikeresés (*lookup*). A keresés bizonyos kritériumoknak megfelelő termékek listáját adja vissza, például *Larry Wall* összes könyvét, vagy olyan könyveket, amelyek címében a *Python* szó szerepel, vagy a *Woody Allen* rendezte filmeket. A kikeresésnek akkor vesszük hasznát, ha ismerjük egy adott termék azonosítóját, az *ASIN*-t (*Amazon Standard ID Number*).

A könyvek *ASIN*-ja megegyezik az *ISBN* azonosítóval (*International Standard Book Number*), más terméktípusok azonosítóját az *Amazon* határozza meg.

Vegyünk egy példát: azt szeretném megtudni, hogy megtalálható-e az *Amazonnál* a *Pragmatic Programmers' book* című könyv a *Ruby on Rails*-hez, és jó lenne tudni az árát is. Mivel egy bizonyos dolgot keresek, érdemes az *ItemLookup* (elem kikeresése) művelet igénybe venni. Ez azt jelenti, hogy tudnom kell a könyv *ISBN*-jét, ami pedig *097669400X*. (Az *ISBN*-t kötőjelek és más írásjelek nélkül kell megadni az *ECS*-nek.) Végül szükség van egy *AccessKeyId*-re (hozzáférésikülcs-azonosító) értékére, ami egy azonosítószám: azt jelzi az *Amazon*nak, melyik fejlesztő próbál hozzáférni a rendszerhez. (Az *AccessKeyId* könnyen

beszerezhető és ingyenes, a részletekért lásd az online forrásokat.)

Az *ECS REST* kérések alap *URL*-je

➤ <http://webservices.amazon.com/onca/xml?Service=AWSECommerceService>.

A művelet, az *AccessKeyId* és az *ItemId* (elem azonosító) megadásához név-érték párokat adunk az *URL*-hez. A név=érték formátumot használjuk, a párokat & jellel választjuk el egymástól. Kombinált *URL*-ünk most így fest:

```
http://webservices.amazon.com/
  ↪ onca/xml?Service=AWSECommerce
  ↪ Service&Operation=ItemLookup&
  ↪ AWSAccessKeyId=XXX&ItemId=
  ↪ 0735619530
```

Ezt egy webböngészőbe bemásolva (az *XXX* helyére egy aktuális *AccessKeyId* értéket kell beírni) egy, az *Amazon* kiszolgálójáról érkező *XML* dokumentumot kell kapunk (*text/xml* tartalomtípussal). A dokumentum egy *ItemLookupResponse* címkével kezdődik, majd két részre válik, egy *OperationRequest*-re és egy *Items*-re. Az előbbi leírja az általunk küldött kérést, tartalmazza böngészőnk *UserAgent* fejlécét, valamint az összes, a szolgáltatásnak átadott argumentumot. Az *Items* az *Amazon* válaszait tartalmazza.

Példaképpen bemutatok egy választ, amit én kaptam egy kérésemre az *Amazon*tól:

```
<ItemLookupResponse>
  <OperationRequest>
    <HTTPHeaders>
      <Header Name=
        ↪ "UserAgent"
        ↪ Value="Mozilla/5.0
        ↪ (Macintosh; U; PPC
        ↪ Mac OS X Mach-O; en-US; rv:1.8)
        ↪ Gecko/20051111 Firefox/1.5"/>
    </HTTPHeaders>
    <RequestId>1NBWT1FHDEH
    ↪ JK2G16CT</RequestId>
    <Arguments>
      <Argument Name=
        ↪ "Operation" Value=
        ↪ "ItemLookup"/>
      <Argument Name=
        ↪ "Service" Value="AWSE
        ↪ CommerceService"/>
      <Argument Name="AWS
        ↪ AccessKeyId" Value=
        ↪ "XXX"/>
```

```

        <Argument Name="ItemId"
value="097669400X"/>
    </Arguments>
    <RequestProcessingTime>
    ↪ 0.00745105743408203</
    ↪ RequestProcessingTime>
    </OperationRequest>
    <Items>
    <Request>
    <IsValid>True</IsValid>
    <ItemLookupRequest>
    <ItemId>097669400X</ItemId>
    </ItemLookupRequest>
    </Request>
    <Item>
        <ASIN>097669400X</ASIN>
        <DetailPageURL>
http://www.amazon.com/exec/
    ↪ obidos/redirect?tag=
ws%26link_code=xm%26camp=2025%
    ↪ 26creative=
165953%26path=http://www.amazon
    ↪ .com/gp/
redirect.html%253fASIN=09766940
    ↪ 0X%2526tag=
ws%2526lcode=xm%2526cID=2025%2
    ↪ 526ccmID=
165953%2526location=/o/ASIN/
097669400X%25253FSubscriptionId
    ↪ =XXX
        </DetailPageURL>
        <ItemAttributes>
        <Author>Dave Thomas</
    ↪ Author>
        <Author>David Hansson</
    ↪ Author>
        <Author>Leon Breedt</
    ↪ Author>
        <Author>Mike Clark</
    ↪ Author>
        <Author>Thomas Fuchs</
    ↪ Author>
        <Author>Andrea Schwarz</
    ↪ Author>
        <ProductGroup>Book</
    ↪ ProductGroup>
        <Title>
Agile web Development
    ↪ with Rails (The
    ↪ Facets of Ruby
    ↪ Series)
        </Title>
        </ItemAttributes>
    </Item>
    </Items>
</ItemLookupResponse>

```

Sok hasznos mező található ebben az XML-ben. Megtudhatjuk belőle, mennyi időt vett igénybe az Amazon-

nak, hogy feldolgozza a kérést (ez esetben 0,008 másodpercet) – ez az információ jól jön, ha alkalmazásunkban hibakeresést végzünk és/vagy mérjük a teljesítményét. A DetailPageURL tartalmazza azt az URL-t, ahova azokat a felhasználóinkat irányíthatjuk, akiknek több információra van szükségük. Olyan adatokat is megkapunk, mint a könyvcím és a szerző(k) neve, így a könyvinformációk remekül megjeleníthetők. Valóban nem okozhat nehézséget, hogy elemezzük ezt az XML-t, részleteit vagy az egészet megjelenítsük egy webes, grafikus, vagy parancssori alkalmazásban. Azt is megtehetjük, hogy az adatok egy részét hozzáadjuk egy nagyobb adatbázis-alkalmazásunkhoz, de győződjünk meg róla, hogy ezzel nem sértjük meg az Amazon adatvédelmi korlátozásait.

Válaszcsoportok

Legyenek bármilyen hasznosak ezek a fenti információk, eredeti kérdésünkre még mindig nem tudjuk a választ, miszerint kapható-e az Amazon-nál a *Pragmatic Programmers'* könyv a *Ruby on Rails*-hez, és mennyibe kerül. Azt tudjuk, hogy kapható, de nem tudjuk, mennyiért. Ez azért van, mert az ECS alapértelmezetten csak kevés adatot ad vissza, mint az fent is látható. Válaszcsoportok meghatározásával azonban testre szabhatók az Amazon válaszai. Minden egyes válaszcsoport megfelel egy vagy több, az ECS válaszban visszaadott adattípusnak. Ahhoz, hogy alapvető árinformációkat szerezzünk egy könyvről, jelezzük, hogy az OfferSummary válaszcsoportot szeretnénk megkapni:

```

http://webservices.amazon.com/
    ↪ onca/xml?Service=AWSECommerce
    ↪ Service&Operation=ItemLookup&
    ↪ AWSAccessKeyId=XXX&ItemId=073
    ↪ 5619530&ResponseGroup=Offer
    ↪ Summary"

```

Az előző, magát a könyvet leíró lista helyett most egy olyat kapunk, amelyben egy adott könyv legalacsonyabb új és régi árai szerepelnek. A fenti lekérésre az XML válasz a következő:

```

<ItemLookupResponse>
<OperationRequest>
<HTTPHeaders>

```

```

<Header Name="UserAgent"
    ↪ Value="Mozilla/5.0
    ↪ (Macintosh; U; PPC Mac
    OS X Mach-O; en-US; rv:1.8)
    ↪ Gecko/20051111 Firefox/1.5"/>
</HTTPHeaders>
<RequestId>0SNXJ8T5V2JA18M8AJQC
</RequestId>
<Arguments>
<Argument Name="ResponseGroup"
    ↪ value="OfferSummary"/>
<Argument Name="Operation"
    ↪ value="ItemLookup"/>
<Argument Name="Service"
    ↪ value="AWSECommerceService"/>
<Argument Name="AWSAccessKeyId"
    ↪ value="XXX"/>
<Argument Name="ItemId"
    ↪ value="097669400X"/>
</Arguments>
<RequestProcessingTime>0.033176
    ↪ 8989562988</RequestProcessing
    ↪ Time>
</OperationRequest>
<Items>
    <Request>
    <IsValid>True</IsValid>
    <ItemLookupRequest>
    <ItemId>097669400X</
    ↪ ItemId>
    <ResponseGroup>Offer
    ↪ Summary</Response
    ↪ Group>
    </ItemLookupRequest>
    </Request>
    <Item>
    <ASIN>097669400X</ASIN>
    <OfferSummary>
    <LowestNewPrice>
    <Amount>2295</Amount>
    <CurrencyCode>USD</
    ↪ CurrencyCode>
    <FormattedPrice>
    ↪ $22.95</Formatted
    ↪ Price>
    </LowestNewPrice>
    <LowestUsedPrice>
    <Amount>2341</Amount>
    <CurrencyCode>USD</
    ↪ CurrencyCode>
    <FormattedPrice>
    ↪ $23.41</Formatted
    ↪ Price>
    </LowestUsedPrice>
    <LowestCollectible
    ↪ Price>
    <Amount>3495</Amount>
    <CurrencyCode>USD</
    ↪ CurrencyCode>
    <FormattedPrice>

```

```

    <Price>34.95</Formatted
    <Price>
  </LowestCollectible
    <Price>
  <TotalNew>41</TotalNew>
  <TotalUsed>12</
    <TotalUsed>
  <TotalCollectible>2</
    <TotalCollectible>
  <TotalRefurbished>0</
    <TotalRefurbished>
  </OfferSummary>
</Item>
</Items>
</ItemLookupResponse>

```

Látható, a válasz eleje ugyanaz, mint az előbb volt. A második fele az <Items> címkék között azonban más, most LowestNewPrice, LowestUsedPrice és LowestCollectiblePrice címkék mutatják mennyiért vehetjük meg a könyvet.

Más válaszcsoportokat is választhatunk, szükség szerint kombinálhatjuk és összeilleszthetjük őket. Kérhetjük például a *Medium* válaszcsoportot, amely nem csak a kérésről és a könyvről ad információt, hanem

a könyvvel kapcsolatos képeket is mutat (több méretben), megtudjuk a könyv méretét és súlyát, valamint elolvashatjuk a szerkesztői ismertetőt. Ha ennél is többet szeretnénk, például más Amazon-vásárlók által írt véleményt a könyvről, vagy a hasonló termékek listáját, kérjük a *Large* válaszcsoportot.

Összefoglalás

Az *Amazon* webszolgáltatások egy olyan eszközt bocsátanak rendelkezésünkre, amellyel termékinformációkat szerezhethetünk egy hatalmas adatbázisban, akár személyes, akár kereskedelmi célra. Továbbá az *ECS* ízelítőt ad abból, hogyan hozhatók létre *REST*-szerű kérések, és az eredmények hogyan elemezhetők. Valamint csakúgy, ahogy a webfejlesztők is sokat tanulhatnak a *HTML*-ről és a *JavaScript*-ről a meglévő oldalak alapján, mi is elleshetjük a webszolgáltatások létrehozásának titkait, csak tanulmányoznunk kell, az *Amazon* hogyan oldotta meg a kérdést. Én magam az *Amazon* válaszcsoport-megvalósítását kedvelem különösen, amely lehetővé teszi, hogy kombináljuk és összeillesztjük a várt

választípusokat, nos, ez olyan dolog, amit én is szeretnék a saját webszolgáltatásomba integrálni. A következő hónapban meg is csináljuk azt, amiről most szó volt, azaz építünk egy saját webszolgáltatást, amely az *Amazon*, valamint egy helyi nyilvános könyvtár adatait használja fel egy személyre szabott könyvkereső rendszer létrehozására.

Linux Journal 2006., 143. szám



Reuven M. Lerner

hosszú ideje web- és adatbázis-tanácsadóként dolgozik, emellett PhD-s oktatástudomány (Learning Sciences)

hallgató a chicagói Northwestern University-n. Chicago külvárosában él feleségével és három gyermekével, köztük Amotz David névre hallgató újszülött kisfiával.

A CIKK FORRÁSAI

www.linuxjournal.com/article/8748



Hungarian
Unix
Portal

A magyar Unix hírek forrása

Linux, Unix, BSD hírek

Cikkek

Értékelések

Technikai segítségnyújtás

Fórumok

Magyar nyelvű dokumentációk

www.hup.hu