

Ajax and Web Services in PHP

Google AdWords API with APIlity

Forum PHP 2006, Paris
November 9, 2006

Thomas Steiner Steiner.Thomas@gmail.com
in cooperation with
Patrick Chanezon chanezon@google.com
Jacob Moon jacobm@google.com



-  Presentation of the speaker and introduction
-  About Google AdWords and the AdWords API
-  Introduction of the Google APIlity Library (live demo)
-  AdWords goes Ajax: APIlity "ajaxified" (live demo)
-  Mashing up Google Checkout and AdWords (live demo)
-  Questions

Presentation of Thomas Steiner



Maintainer of the **APIlity Library**
and the **APIlitAx project**

Steiner.Thomas@gmail.com
<http://blog.tomayac.de>

Google engineering intern in summer 2006

French/German double degree MS CS grad student

– ENSIMAG (Grenoble, France)



– University of Karlsruhe (Germany)



Introduction to Google AdWords



How does AdWords work?

- **You create your ads**

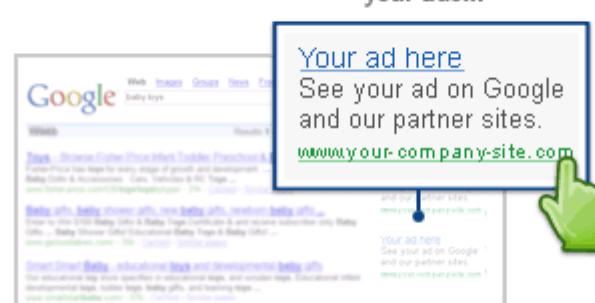
You create ads and choose keywords, which are words or phrases related to your business

- **Your ads appear on Google**

When people search on Google using one of your keywords, your ad may appear next to the search results

- **People can simply click** your ad to make a purchase or learn more about you

Your ads appear beside related search results...



People click your ads...

...And connect to your business



Google's AdWords API

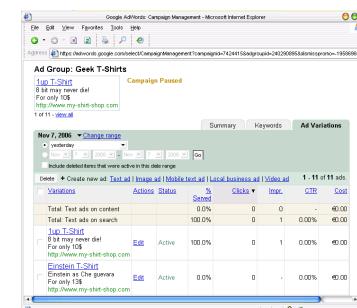


Why does AdWords require an API?

API = Application Programming Interface, permits two or more systems to talk efficiently to each other



AdWords API



Web Front End

The API defines a programmatic way to access AdWords

Google's AdWords API (cont'd)



What is the AdWords API good for?

- Lets developers design computer programs that interact directly with the AdWords platform.
- Allows automation of common tasks such as:
 - Generating automatic keyword, ad text, and custom **reports**
 - **Integrating AdWords** with databases or inventory systems
 - Developing **additional applications**

Who can use the AdWords API?

- All advertisers can use the API
- Simply sign up for a **My Client Center** account



Google's AdWords API (cont'd)



How is API usage administrated?

The screenshot shows the 'AdWords API Center' section of the Google AdWords interface. It includes sections for 'API Usage' (showing 0 quota units used and 250,000 remaining), 'Rate and Budget' (set at \$0.25/1000 quota units monthly budget of \$1.00), 'Your Developer Token' (a randomly generated string of lowercase letters: abcdefghijklmnopqrstuvwxyz), 'Your Application Tokens' (another randomly generated string: abcdefghijklmnopqrstuvwxyz), and 'Developer Details' (company information: Independent software developer, Google, http://google.com, 501+ employees, chanezon@gmail.com).

- Each API user is assigned a unique *developer token* which is linked to the user's My Client Center account
- Quota costs *\$0.25/1000 quota units*, billed to a developer token
- *Application token* is specific to the application you develop
- Users can access information about their *API usage* via the AdWords API Center and
 - Generate, view, or reset their developer token
 - Check their quota allocation and quota units consumed

Google's AdWords API (cont'd)



Where can I get started?

- Using the Developer Website, developers can find valuable resources to help them create applications
- Designed to be a do-it-yourself program

The screenshot shows the AdWords API developer website. At the top left is the Google logo. The main header reads "AdWords API". The left sidebar contains links for "AdWords API Home", "Overview", "Documentation", "Support", and "Toolbox". The main content area has several sections: "Introduction" (explains the API's purpose), "Limitless Functionality" (lists possibilities like generating reports and integrating with databases), "Multi-Language Compatibility" (notes support for Java, PHP, Perl, C#, and XML), "Simple and Speedy Signup" (describes the process), and "Ready to sign up? Click here.". A sidebar on the right includes "Sign in to your AdWords API Center", "Sample Code" (links to Java, .NET, Perl, PHP, and XML), "Support" (links to AdWords API FAQ and Ask Other Developers), and a note about documentation being in English only.

Key components:

- Developer's Guide
- Developer Forum
- Frequently Asked Questions
- Sample Code
- Blog

<http://www.google.com/apis/adwords>

Google's AdWords API (cont'd)



What does API usage cost?

- Usage is measured in quota
- Each unit costs \$0.25

Type of Operation	Quota Unit(s)
Keyword status	1 unit
Change keyword bid	10 units
Traffic estimator	25 units
Add keyword	50 units
Add creative	250 units
Pull report	1000 units

- This is a simplified version of the actual rate sheet, which lists all available API services
- However, the numbers reflected here are accurate and represent all the changes that are being made

How do I sign up for API usage?

Production



- Go to <http://www.google.com/apis/adwords>
- Create your **MCC (My Client Center) account** using your email address and password
- Developer and application tokens will be generated
- Set a limit for API quota: you pay for quota
- Visit the API website to view more information including documentation, sample code, and the Developer Forum

Sandbox



- For development create a sandbox account
- **No quota cost** there

Google's AdWords API (cont'd)

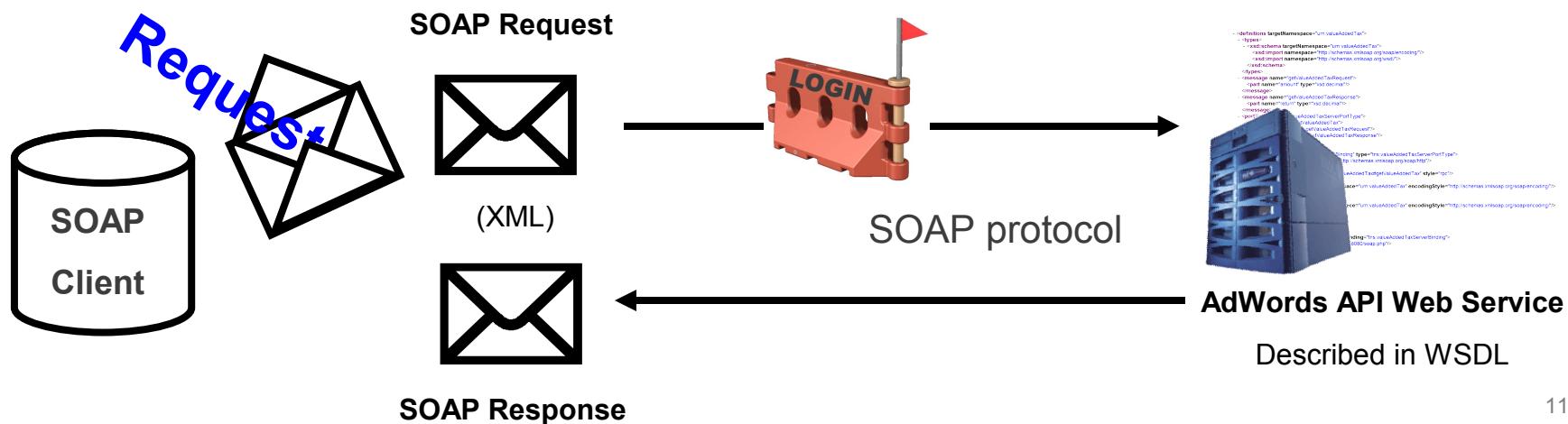


What do I need in order to use the AdWords API?

- **SOAP-based API** → multi-language compatibility
- Supported languages: Java, .NET, Perl, PHP, Python, OCaml, Ruby, XML, and more...

What is SOAP?

- Protocol for exchanging **XML-based messages** over a computer network, normally using HTTP



What goes actually on the wire?

```
<?xml version="1.0" encoding="utf-8" ?>
<env:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<env:Header>
    <developerToken env:mustUnderstand="0">your_dev_token
    </developerToken >
    <applicationToken env:mustUnderstand="0">your_app_token
    </applicationToken >
    <useragent env:mustUnderstand="0">P@ playing with the API from PHP
    </useragent>
    <password env:mustUnderstand="0">your_password
    </password>
    <email env:mustUnderstand="0">toto@example.com
    </email>
</env:Header>
<env:Body>
    <estimateKeywordList xmlns="https://adwords.google.com/api/adwords/v6">
        <keywordRequests>
            <maxCpc>50000</maxCpc>
            <text>flowers</text>
            <type>Broad</type>
        </keywordRequests>
    </estimateKeywordList>
</env:Body>
</env:Envelope>
```

What's in the SOAP header?

```
<env:Header>
  <developerToken env:mustUnderstand="0">your_dev_token
  </developerToken >
  <applicationToken env:mustUnderstand="0">your_app_token
  </applicationToken >
  <useragent env:mustUnderstand="0">P@ playing with the API
  </useragent>
  <password env:mustUnderstand="0">your_password
  </password>
  <email env:mustUnderstand="0">toto@example.com
  </email>
</env:Header>
```

How about the SOAP body?

- Schema defined in the WSDL file for the service
- Namespaced with version specific URL

`https://adwords.google.com/api/adwords/vN`

```
<env:Body>
  <estimateKeywordList
    xmlns="https://adwords.google.com/api/adwords/v6">
    <keywordRequests>
      <maxCpc>50000</maxCpc>
      <text>flowers</text>
      <type>Broad</type>
    </keywordRequests>
  </estimateKeywordList>
</env:Body>
```

AdWords SOAP details (cont'd)



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
    <soapenv:Header>  
        <responseTime soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"  
            soapenv:mustUnderstand="0"  
            xmlns="https://adwords.google.com/api/adwords/v6">1999</responseTime>  
        <operations soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"  
            soapenv:mustUnderstand="0"  
            xmlns="https://adwords.google.com/api/adwords/v6">1</operations>  
        <units soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"  
            soapenv:mustUnderstand="0"  
            xmlns="https://adwords.google.com/api/adwords/v6">25</units>  
        <requestId soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"  
            soapenv:mustUnderstand="0"  
            xmlns="https://adwords.google.com/api/adwords/v6">e701e5123f[...]</requestId>  
    </soapenv:Header>  
    <soapenv:Body>  
        <estimateKeywordListResponse xmlns="https://adwords.google.com/api/adwords/v6">  
            <estimateKeywordListReturn>  
                <id>-1</id>  
                <lowerAvgPosition>7.0</lowerAvgPosition>  
                <lowerClicksPerDay>111.112045</lowerClicksPerDay>  
                <lowerCpc>50000</lowerCpc>  
                <upperAvgPosition>10.0</upperAvgPosition>  
                <upperClicksPerDay>167.89006</upperClicksPerDay>  
                <upperCpc>50000</upperCpc>  
            </estimateKeywordListReturn>  
        </estimateKeywordListResponse>  
    </soapenv:Body>  
</soapenv:Envelope>
```

AdWords SOAP details (cont'd)



What's in the SOAP response body?

```
<soapenv:Body>
  <estimateKeywordListResponse
    xmlns="https://adwords.google.com/api/adwords/v6">
    <estimateKeywordListReturn>
      <id>-1</id>
      <lowerAvgPosition>7.0</lowerAvgPosition>
      <lowerClicksPerDay>111.112045</lowerClicksPerDay>
      <lowerCpc>50000</lowerCpc>
      <upperAvgPosition>10.0</upperAvgPosition>
      <upperClicksPerDay>167.89006</upperClicksPerDay>
      <upperCpc>50000</upperCpc>
    </estimateKeywordListReturn>
  </estimateKeywordListResponse>
</soapenv:Body>
```

How about error handling?

- SOAP fault elements
- Include AdWords specific fields: code, message, trigger

```
<soapenv:Fault>
  <faultcode>soapenv:Server.userException</faultcode>
  <faultstring>Invalid client email specified.</faultstring>
  <detail>
    <ns1:code xmlns:ns1="https://.../v6">9</ns1:code>
    <ns2:message xmlns:ns2="https://.../v6">
      Invalid client email specified.</ns2:message>
    <ns3:trigger xmlns:ns3="https://.../v4">toto</ns3:trigger>
  </detail>
</soapenv:Fault>
```

I don't care about SOAP. I speak {Java|PHP|Perl|...}



No, thanks!

- External developers and Google offer **client libraries** that provide an abstraction from SOAP
 - Java Client Library (Java)
 - Adwords4r (Ruby)
 - Merjis AdWords Toolkit (OCaml)
 - ...

This presentation is about APIlity [əpɪlɪtɪ]

- Like "ability" but with 'p', for **PHP** and the **ability** to do whatever you want with the **API**.

Tell me more about APIlity's features!

- Fully **object oriented**
- Runs in both **PHP4**, and **PHP5**
- **Consistent** in usage, all objects have adequate methods
- Released under the very business-friendly **BSD licence**
- **Open source** project - everyone is encouraged to modify, read or learn from the sources



Which credo has guided the development of APIlity?

- Larry Wall: "*Easy things should be easy and hard things should be possible*"

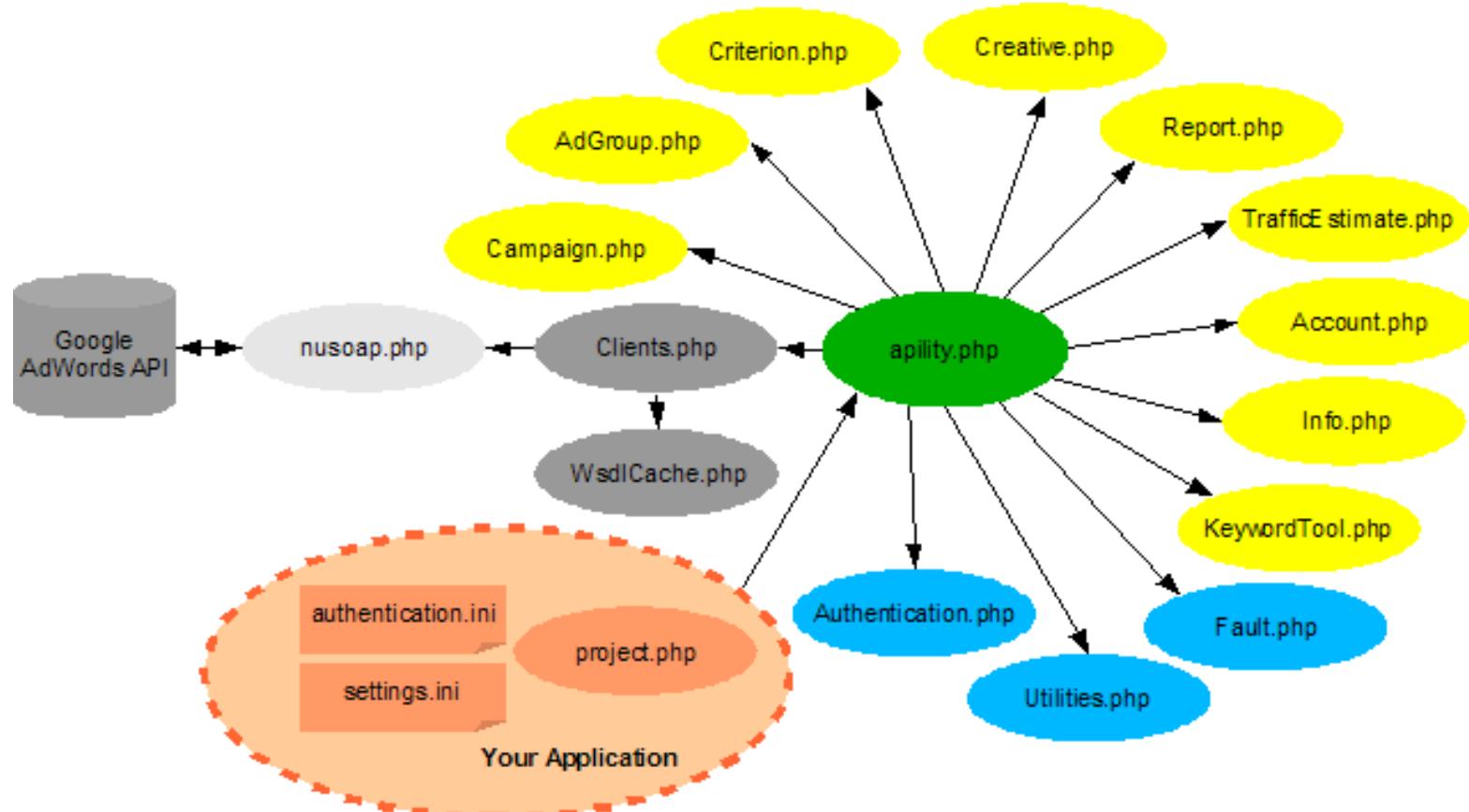
→ <http://google-apility.sourceforge.net>



Google APIliby Library (cont'd)



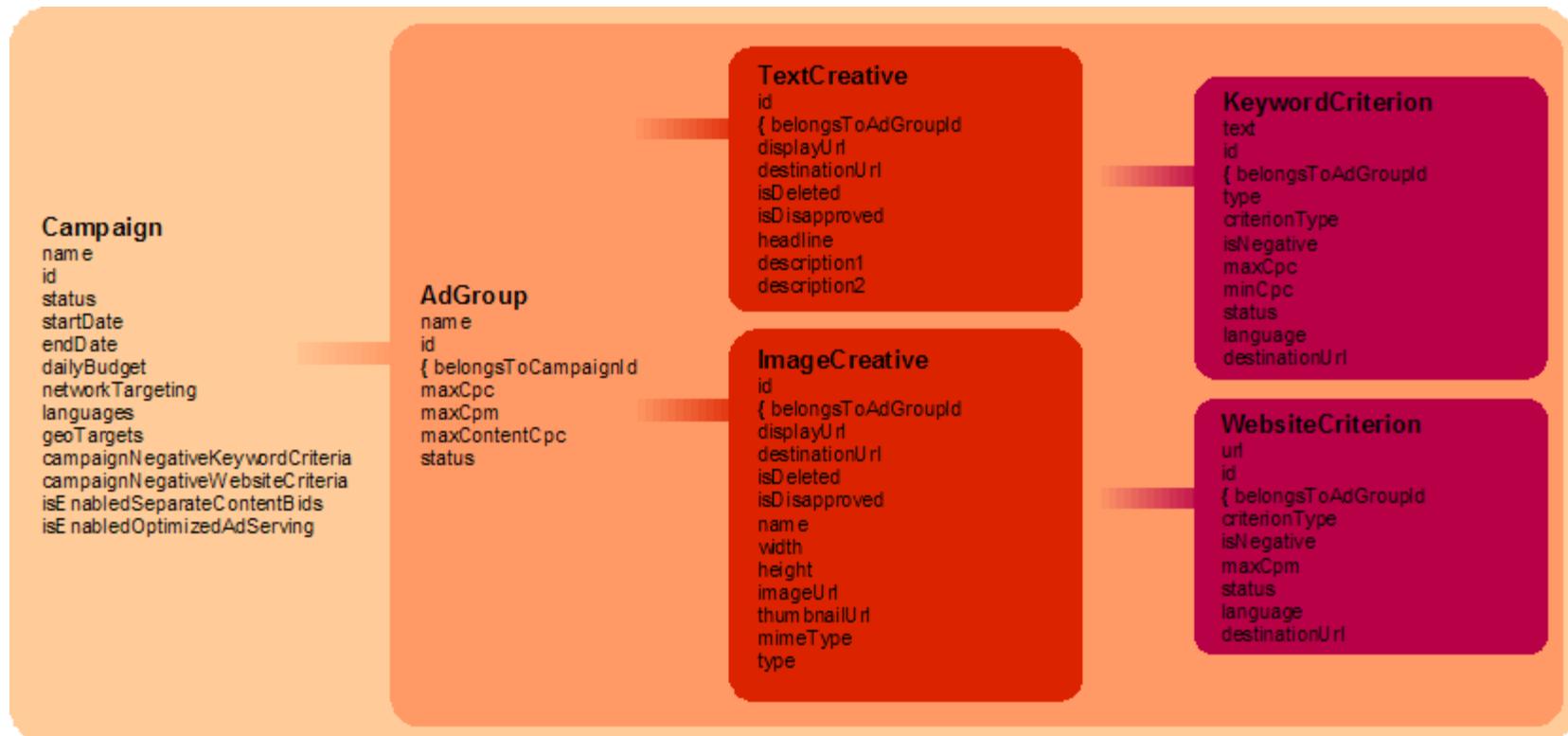
APIliby's class architecture



Google APIliby Library (cont'd)



APIliby's object hierarchy



Object oriented design:

```
- $adGroupObject = createAdGroupObject(123456789) ;  
$adGroupObject->setName ("newName") ;  
$adGroupObject->setMaxCpc (0.65) ;  
echo "MaxCPC: ".$adGroupObject->getMaxCpc ()
```

Principle of the least surprise → consistency

- Setting object statuses
 - \$adGroupObject->setStatus ("Paused") ;
 - \$campaignObject->setStatus ("Paused") ;
- Removing objects
 - removeAdGroup (\$adGroupObject) ;
 - removeCampaign (\$campaignObject) ;
 - removeCriterion (\$criterionObject) ;

Programming samples:

- Hello World
 - ```
require_once('apility.php');
print_r(getAllCampaigns());
```
- Adding a new keyword: `addKeywordCriterion()`
  - Parameters:
    - `String text`
    - `Integer belongsToAdGroupId`
    - `String type`
    - `Boolean isNegative`
    - `Integer maxCpc`
    - `String language`
    - `String destinationUrl`
  - ```
$criterion = addKeywordCriterion("Forum PHP",
123456789, "Exact", false, 0.65, "fr",
"http://www.google.fr");
```

Having fun with APIlity (cont'd)



APIlity live demo

APIlity allows data export in XML format

- ```
$campaignObject = createCampaignObject(123456789);
echo ($campaignObject->toXml());
```
- XML? → Ajax

## Idea



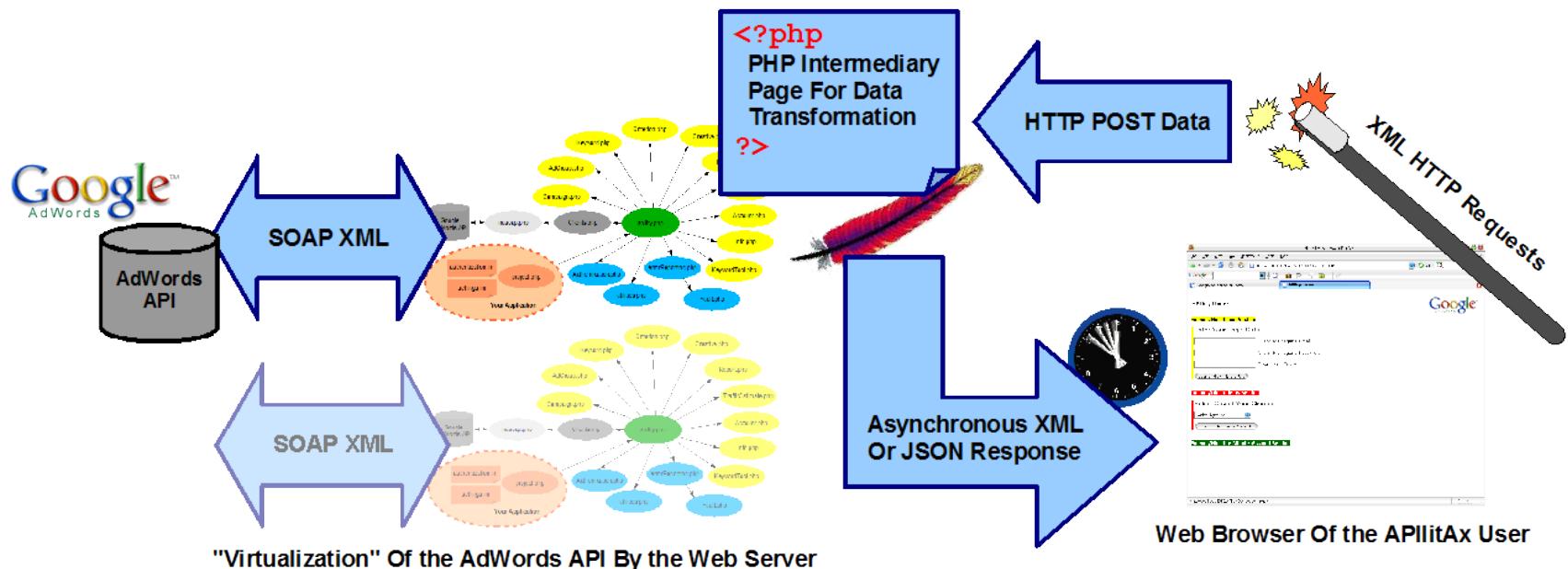
- Have a complex web application do the **GUI** stuff (JavaScript)
- Use APIlity calls for the AdWords API communication (PHP)
- Use XML for the communication between the two layers
- Call the project APIlity + Ajax = **APIlitAx**

# Having fun with APIlity (cont'd)



## Principle

- JavaScript XML HTTP requests from a web application access intermediate local PHP files communicating via HTTP POST
- PHP files make the APIlity API calls, running on the local server
- Asynchronous responses are translated back into JavaScript



# Having fun with APIlity (cont'd)



What does it look like?

- Client Select section:

→ <http://apilitax.sourceforge.net>



# Having fun with APIlity (cont'd)

Google™

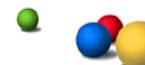
What does it look like?

- Account Structure section:

The screenshot shows the APIlity interface for Google AdWords. At the top, there's a header bar with the APIlity logo, the text "An Ajax GUI for Google AdWords", the Google AdWords logo, and the text "...powered by APIlity". Below the header is a navigation bar with "Login", "Client Select", and "Account Structure". The main content area is titled "Account Structure Section". It displays two campaigns: "cou" and "RegMetCit". Each campaign has fields for Campaign Name, DailyBudget, StartDate, NetworkTargeting, GeoTargets, and Languages. The "cou" campaign also includes fields for Status, EndDate, and AdGroups. The "RegMetCit" campaign includes fields for Details, Detach, and SeparateContentBids.

| Campaign Name     | DailyBudget | StartDate                | NetworkTargeting                                                                                                                                 | GeoTargets                                                                                  | Languages |
|-------------------|-------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|-----------|
| cou<br>5895       | 50          | 2006-10-19<br>2011-01-01 | <input checked="" type="checkbox"/> ContentNetwork<br><input checked="" type="checkbox"/> GoogleSearch<br><input type="checkbox"/> SearchNetwork | <a href="#">Modify...</a><br>Countries<br>FR<br>DE<br>+1 other                              | de<br>fr  |
| RegMetCit<br>5899 | 50          | 2006-10-19<br>2011-01-01 | <input checked="" type="checkbox"/> ContentNetwork<br><input checked="" type="checkbox"/> GoogleSearch<br><input type="checkbox"/> SearchNetwork | <a href="#">Modify...</a><br>Regions Metros<br>US-CA 527<br>US-HI 548<br>+1 other +2 others | de<br>fr  |

[APIlity](#) is an open source project released under the [BSD license](#). It is hosted on [SourceForge](#) and provided "as is" without any warranty. Graphics based on Everaldo's [Crystal icon theme](#).



→ <http://apilitax.sourceforge.net>



# Having fun with APIlity (cont'd)



APIlitAx live demo

# Google Checkout and AdWords



## What is Google Checkout?

- Many shopping carts are cancelled during checkout process
- Google Checkout is an alternative checkout process that provides a **fast, secure way to buy and sell online**

The screenshot shows a Google search results page for the query "black handbags". The results are displayed under the "Web" category. The top result is a sponsored link from Coach.com. Below it is a snippet from All the HandBags.com. Further down are links from Sophie Bags, RetailCo.com (with a large graphic overlay), JCPenney, and Alternative Beauty. To the right of the main search results, there are two columns of sponsored links: one for Designer Handbags Online and another for Black Purses.

Google™ Web Images Groups News Froogle Local more »

black handbags Search Advanced Search Preferences

Results 1 - 10 of about 18,500,000 for [black handbags](#) (0.24 seconds)

**Web**

[Handbags at Coach.com](#) Sponsored Links  
www.Coach.com Find New & Stylish Coach Designs - Online Now at the Official Site

[All the HandBags](#) Sponsored Links  
Handbags

[Find the Perfect Black Handbag](#)  
Fabulous & Unusual Styles. Great Customer Service & Free Shipping!

[Black Purse](#) Start Here.  
Sophie Bags ha ... Sophie Bags www.sophiebag

[Black Hand](#)  
My name is Stacey Leigh DIAZ, I am 27, and live in Greebo NCxx. Like most girls, I have a deep obsession for handbags, which I change as often as my shoes. ...  
www.blackhandbags.com.au/ - 7k - [Cached](#) - [Similar pages](#)

[black handbags shop by color : women's : JCPenney](#)  
Women's, Handbags Shop By Color, Black ... 'Call Me' Leather Handbag · St. John's Bay Woven Double Handle Shopper · St. John's Bay Woven Double Handle ...  
www.jcpenney.com/products/C033523.jsp - 46k - [Cached](#) - [Similar pages](#)

[Handbags Purses - Designer Handbags Discount](#)  
Black Handbags, leather handbags, designer purses and designer wallets inspired by top designers like Louis Vuitton, Prada, Coach, Gucci and more! ...  
www.alternative-beauty.com/handbags/black.html - 30k - [Cached](#) - [Similar pages](#)

[Designer Handbags Online](#)  
All Major Brands. Huge Selection. Free Shipping. Order Now!  
Zappos.Com

[Black Purses](#)  
Compare Prices on a Wide Variety of Apparel and Save Money.  
PriceGrabber.com

[Find the Perfect Black Handbag](#)  
Fabulous & Unusual Styles. Great Customer Service & Free Shipping!  
www.RetailCo.com

[Women's Handbags](#)  
Spring Sale: Up To 50% Off Select Styles and \$5 Standard Shipping.  
www.AnnTaylorLoft.com

[Find the Perfect Handbag](#)  
Fabulous & Unusual Styles. Great Customer Service & Free Shipping!  
www.eBags.com

# Google Checkout and AdWords (cont'd)



AdWords and Google Checkout work together to

- increase sales
- lower costs.

2 Get more traffic with the  
Google Checkout badge

1 Promote your business  
with AdWords

3 Get higher sales conversions  
by offering a streamlined  
checkout process

4 Lower your costs with free  
transaction processing



Checkout APIlitAx...

## What does it look like?

- Vendor Side

Screenshot of the Checkout APIitAx vendor side interface:

The browser title bar shows: http://localhost:8080/checkout\_apilitax/checkout\_apilitax\_1.6.0/index\_vendor.php#

The page header includes the Google Checkout logo and "Google ...powered by APIitAx".

A banner at the top says "Vendor Side (Your Online Store Back-End)".

Section "Online Store Inventory as retrieved from Database:" displays a grid of items:

| Image | Item Name          | Item Description         | Unit Price | In Stock | ID | DB Access                   | AdWords Access                 |
|-------|--------------------|--------------------------|------------|----------|----|-----------------------------|--------------------------------|
|       | Iup T-Shirt        | 8 bit may never die!     | 10\$       | 50       | 1  | <a href="#">Update Item</a> | <a href="#">Create AdWords</a> |
|       | Einstein T-Shirt   | Einstein as Che          | 13\$       | 130      | 2  | <a href="#">Update Item</a> | <a href="#">Create AdWords</a> |
|       | Major League Gamer | For hardcore players.    | 13\$       | 500      | 3  | <a href="#">Update Item</a> | <a href="#">Create AdWords</a> |
|       | The ESC T-Shirt    | Hit and then run.        | 12\$       | 40       | 4  | <a href="#">Update Item</a> | <a href="#">Create AdWords</a> |
|       | WASD T-Shirt       | Gamers know what this is | 12\$       | 40       | 5  | <a href="#">Update Item</a> | <a href="#">Create AdWords</a> |
|       | Perl T-Shirt       | Obfuscated Perl Code.    | 10\$       | 10       | 6  | <a href="#">Update Item</a> | <a href="#">Create AdWords</a> |

At the bottom of the inventory grid is the URL: <http://www.my-shirt-shop.com>.

A modal dialog titled "Add Creative" is open on the right, showing fields for Headline, Description 1, Description 2, Display Url, and Destination Url, along with "Add Creative" and "Cancel" buttons.

[Checkout APIitAx](#) is an open source project released under the [BSD license](#). It is hosted on [SourceForge](#) and provided "as is" without any warranty. Graphics based on Everaldo's [Crystal icon theme](#).



## What does it look like?

– Vendor Side

The screenshot shows a Mozilla Firefox browser window with the title "Checkout APIlitAx - Google Checkout in Combination with Google AdWords - Mozilla Firefox". The URL in the address bar is [http://localhost:8080/checkout\\_apilitax/checkout\\_apilitax\\_1.6.0/index\\_vendor.php#](http://localhost:8080/checkout_apilitax/checkout_apilitax_1.6.0/index_vendor.php#). The page displays a list of items from a shirt shop, each with a "Link Creative" button. A modal dialog titled "Link Creative to an Item" is open, showing a list of items to link to, with "1up T-Shirt" selected.

Items listed on the left:

- 502186755: Tup T-Shirt, 8 bit may never die!, For only 10\$, <http://www.my-shirt-shop.com> [http://www.my-shirt-shop.com?shirtId=1]
- 502120455: Tup T-Shirt, 8 bit may never die!, For only 10\$, <http://www.my-shirt-shop.com> [http://www.my-shirt-shop.com?shirtId=1]
- 501680475: Tup T-Shirt, 8 bit may never die!, For only 10\$, <http://www.my-shirt-shop.com> [http://www.my-shirt-shop.com?shirtId=1]
- 501679855: Major League Gamer, For hardcore players., For only 13\$, <http://www.my-shirt-shop.com> [http://www.my-shirt-shop.com?shirtId=3]
- 501642555: Tup T-Shirt, 8 bit may never die!, For only 10\$, <http://www.my-shirt-shop.com> [http://www.my-shirt-shop.com?shirtId=1]
- 501642135: Einstein T-Shirt, Einstein as Che Guevara, For only 13\$, <http://www.my-shirt-shop.com> [http://www.my-shirt-shop.com?shirtId=2]
- 501642015: (empty)

Link Creative to an Item dialog:

Link this Creative:  
1up T-Shirt  
8 bit may never die!  
For only 10\$  
<http://www.my-shirt-shop.com>

to this Item:  
 1up T-Shirt

## What does it look like?

- Buyer side

Screenshot of the Checkout APIlitAx application showing the buyer side interface.

**Select Items and Number to purchase:**

| Image                 | Item Name          | Item Description            | Unit Price     | Quantity                           |
|-----------------------|--------------------|-----------------------------|----------------|------------------------------------|
|                       | 1up T-Shirt        | 8 bit may never die!        | 10\$           | 0 <input type="button" value="+"/> |
|                       | Einstein T-Shirt   | Einstein as Che             | 13\$           | 0 <input type="button" value="+"/> |
|                       | Major League Gamer | For hardcore players.       | 13\$           | 0 <input type="button" value="+"/> |
|                       | The ESC T-Shirt    | Hit and then run.           | 12\$           | 2 <input type="button" value="+"/> |
|                       | WASD T-Shirt       | Gamers know what this means | 12\$           | 3 <input type="button" value="+"/> |
|                       | Perl T-Shirt       | Obfuscated Perl Code.       | 10\$           | 0 <input type="button" value="+"/> |
| <b>Overall Price:</b> |                    | <b>60\$</b>                 | <b>5 Items</b> |                                    |

**Hidden Shopping Cart Section**

```

HTTP POST Data (Plaintext)
<itemName=The%20ESC%20T-Shirt<itemDescription=Hit%20and%20then%20un.<

Shopping Cart (XML)
<?xml version="1.0" encoding="UTF-8"?>
<checkout-shopping-cart xmlns="http://checkout.google.com/schema/2">
 <shopping-cart>
 <items>
 <item>
 <item-name>The ESC T-Shirt</item-name>
 <item-description>Hit and then run.</item-description>
 <unit-price currency="USD">12</unit-price>
 <quantity>2</quantity>
 </item>
 <item>
 <item-name>WASD T-Shirt</item-name>
 <item-description>Gamers know what this means</item-description>
 </item>
 </items>
 </shopping-cart>
</checkout-shopping-cart>

```

```

Shopping Cart (Base64-encoded XML)
PD94bWwgdmVyc2lvbj0iMS4wIiBsb3MvZGluZz01VVRLTgiPz4KPGNoZWnrb3V0LXNob3BwaW5nLWnhcnQ=

```

```

Shopping Cart Signature (Base64-encoded HMAC_SHA1 Signature)
vpd1gOJN6Y1cd6phAfx9dqtxFq4=

```

**This Part is Normally Invisible**



[Checkout APIlitAx](#) is an open source project released under the [BSD license](#). It is hosted on [SourceForge](#) and provided "as is" without any warranty. Graphics based on Everaldo's [Crystal icon theme](#).

Checkout APIlitAx live demo

## Questions



Right here, right now

or

[Steiner.Thomas@gmail.com](mailto:Steiner.Thomas@gmail.com)