

Introducing Carousel API

by Noesis Srl
<http://www.noesis-research.com>

Version 1.0



CAROUSEL

Table of Contents

Introduction.....3

The SOAP API Interface.....3

Getting Started Using the Carousel Service.....4

Carousel API

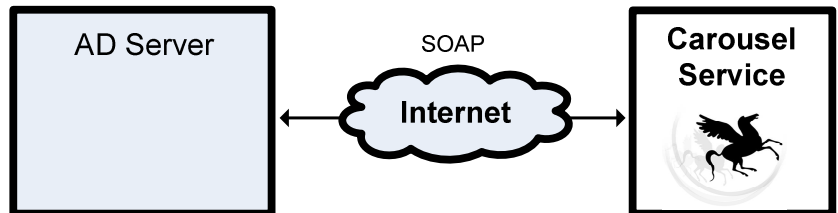
CAROUSEL



Introduction

This document describes the Carousel Optimization Engine Architecture and describes how to access the Carousel API from within an existing AD Server.

The Carousel Service adopts a *Service Oriented Architecture* (SOA) and lets developers design computer programs that interact directly with the Carousel engine. Doing that, ad servers and third-parties can integrate and use the Carousel optimization engine and algorithms.



The SOAP API Interface

The core messaging technology for the Carousel API is Simple Object Access Protocol (SOAP), which is an XML and HTTP-based protocol with wide support in the industry. The Carousel API is exposed via a WS-I Profile 1.1 web service, called Carousel Service. The Carousel API SOAP interface supports a growing number of languages - including Java, .NET, Perl, PHP, Python, OCAML, Ruby and XML.

To access the Carousel engine programmatically, you build web service clients that use the Carousel Service. Typically, to use the Carousel Service, you would download a toolkit that knows how to interpret WSDL files and how to encode and decode XML request and response messages. When a Carousel API Web Service receives a request, it sends back the response as an XML message. The web service toolkits know how to parse the response and return a data structure or object back to the caller, as appropriate for the language.

The exact toolkit you would use depends on the language you are using. Some toolkits do more for you or work better than others. If you use a SOAP toolkit, you should rarely have to write XML code to use the Carousel Service, since the toolkits handle the XML generation.

To learn more about SOAP, see the SOAP Tutorial at W3Schools: <http://www.w3schools.com/soap/default.asp>



Getting Started Using the Carousel Service

To use the Carousel API, write a client program in a language of your choice (such as Java, Perl, Python, C, C++, PHP). Write your client program to send a request to one of the Carousel Service methods, such as the *CreateCampaign* or the *GetMatchingRules*. The relevant service will process the request and send back a response, which your client program needs to parse.

Since the Carousel API runs as a web service, there is nothing that you need to install related to the Carousel API. The only software you need to install to use the Carousel API is the software for the language and toolkit that you will be using to write your client programs. For example, if you intend to write your client programs in Java, you will need to install Java and also a SOAP toolkit such as Axis.

WSDL: The operations provided by a web service are defined in a WSDL (web services definition language) file which is posted on a website. To connect to a web service, you need to know the URL for the WSDL. The Carousel Service WSDL is available at:

<http://67.228.14.85:8099/v1/CarouselService.asmx>

AD Server Integration

This section illustrates the integration scenarios between the Carousel Technology and an existing AD Server System.

A campaign optimization process is split into 2 phases:

1. **Sites and Campaign configuration:** the first step, before the campaign begins. You have to call some web service methods in order to configure your campaigns and to let Carousel be aware of your scenario. Carousel must know the budget, the advertisements to display, the target sites, the metric to optimize (e.g. CTR), and so on.
2. **Campaign optimization:** when the campaign is configured and started, the AD Server must receive and follow Carousel suggestions. Moreover, the AD Server must periodically send Carousel the feedback (clicks and impressions over the pages).

Let's suppose you have successfully configured the sites and the campaign to optimize (this topic is covered in the *Carousel API Developer's Guide*).



When the campaign is started, the interaction between the AD Server software and the Carousel Service is the following:

1. The AD Server calls the *GetMatchingRules* method or the *GetPlanningRules* method, asking for the updated rules.
2. The AD Server receives a message with the new rules.
3. The AD Server picks the new rules from the message.
4. The AD Server replaces the old rules with the new ones.
5. After some time (e.g. 5 minutes), the AD Server calls the *SetFeedback* method, containing the feedback statistics (e.g. clicks and impressions). The feedback lets Carousel learn and update the rules.
6. It repeats the cycle.

When waiting for a message from Carousel, the AD Server continues to deliver banners using the currently available matching rules.

