

affilinet GmbH

affilinet

Advertiser

Web Services

Technical Documentation

© 2011 affilinet GmbH
18/01/2012

Contents

I.	INTRODUCTION.....	4
II.	GETTING ACCESS.....	4
III.	TOOLBOX WEB SERVICE	4
1.	GetChannels	5
	1. Parameters.....	5
	2. Return Types	5
2.	GetCreativesPerType	6
	1. Parameters.....	6
	2. Return Types.....	6
3.	GetPublisherSegments	7
	1. Parameters.....	7
	2. Return Types	7
4.	GetRateList	8
	1. Parameters.....	8
	2. Return Types	8
IV.	STATISTICS WEB SERVICE.....	9
1.	GetStatisticsPerPublisher	10
	1. Parameters.....	10
	2. Return Types	14
V.	ORDERS WEB SERVICE.....	19
1.	GetTransactions - DEPRECATED	20
	1. Parameters.....	20
	2. Return Types	23
2.	CreateTransaction.....	26
	1. Parameters.....	26
	2. Return Types	29
3.	UpdateTransaction	30
	1. Parameters.....	30
	2. Return Types	31

4.	GetOrders	32
1.	Parameters.....	32
2.	Return Types.....	34
5.	GetBasketItems	38
1.	Parameters.....	38
2.	Return Types.....	39
6.	UpdateBasket	42
1.	Parameters.....	42
2.	Return Types.....	42
7.	UpdateBasketItem	43
1.	Parameters.....	43
2.	Return Types.....	44
8.	CreateBasket	45
1.	Parameters.....	45
2.	Return Types.....	47
VI.	SAMPLE CODE	49
1.	Java	50
1.	UsernameTokenHandler.java	50
2.	UserPasswordInfo.java	51
3.	Main.java	51
2.	C#	56
3.	PHP	58

I. Introduction

The Advertiser Web Services allow affilinet advertisers to access their program data without logging in to their account on the affilinet website. That data is returned as XML, which can then be processed by external applications.

All parameters marked with '*' in this documentation are obligatory fields and must be filled for each web service request. Boolean-type parameters must be stated as 'true' or 'false'.

Please note that for performance reasons, all our web services deliver results no bigger than 20 MB. If the XML serialized result is bigger than that, the web service will throw an exception.

II. Getting Access

In order to get access to the web services, please log in to your advertiser account at <http://www.affili.net> – then go to “Account” > “Account Data”. There you can activate web services for your account, create users for web service interaction and reset passwords if necessary.

Submit these authentication credentials in the SOAP header of each web service request for authentication. Code examples in Java, C# and PHP are available [at the end](#) of this documentation.

III. Toolbox Web Service

This web service contains a collection of methods, which can be used in various contexts and are not tied to a certain use case.

<https://advertiser-webservices.affili.net/V4/WSDLFactory/Toolbox.wsdl>

1. GetChannels

Programs can organize their publishers in two separate channel groups. Each channel group can have multiple channels. This method provides you with the channels, which have been defined for this program.

1. Parameters

GetChannelsRequest

Parameter	Type	Description
ChannelGroup *	Enum	In this parameter, you can define, of which channel group you wish to receive the single channels. Possible values are: <ul style="list-style-type: none"> • ChannelGroup1 • ChannelGroup2
ProgramId *	Integer	The ID of the requesting advertiser

2. Return Types

GetChannelsResponse

Property	Type	Description
ChannelCollection	Array of Channel	List of all channels of the requested channel group

Channel

Property	Type	Description
ChannelId	Integer	The ID of the found channel
ChannelName	String	The name of the found channel

2. GetCreativesPerType

Several web services expect creative identifiers as an input parameter. This method provides you with the necessary input.

1. Parameters

GetCreativesPerTypeRequest

Parameter	Type	Description
CreativeType *	Enum	The basic type of the requested creatives. Possible values are: <ul style="list-style-type: none"> • Text • Banner • HTML
ProgramId *	Integer	The ID of the requesting advertiser

2. Return Types

GetCreativesPerTypeResponse

Property	Type	Description
CreativeCollection	Array of Creative	List of all creatives of the requested creative type

Creative

Property	Type	Description
CreativeNumber	Integer	The Id of the found creative. Please note that the combination of ProgramId, CreativeType and CreativeNumber form a unique identifier.
CreativeName	String	The name of the found creative

3. GetPublisherSegments

Programs can organize their publishers freely in any number of publisher segments for various uses, e.g. statistics, business models, mailings, etc... Several web services expect publisher segment identifiers as an input parameter. This method provides you with these identifiers.

1. Parameters

GetPublisherSegmentsRequest

Parameter	Type	Description
ProgramId *	Integer	The ID of the requesting advertiser

2. Return Types

GetPublisherSegmentsResponse

Property	Type	Description
PublisherSegmentCollection	Array of PublisherSegment	List of all publisher segments which the requesting program has defined

PublisherSegment

Property	Type	Description
SegmentId	Integer	The ID of the found publisher segment
SegmentName	String	The name of the found publisher segment
Description	String	A description on the found publisher segment

4. GetRateList

Several web services expect rate identifiers as an input parameter. This method provides you with the necessary input.

1. Parameters

GetRateListRequest

Parameter	Type	Description
ProgramId *	Integer	The ID of the requesting advertiser

2. Return Types

GetRateListResponse

Property	Type	Description
RateCollection	Array of Rate	List of all rates which the requesting program has defined

Rate

Property	Type	Description
RateNumber	Integer	The number of the found rate – the combination of rate mode, rate number and the program ID is a unique identifier.
RateMode	Enum	The commission mode which is used to remunerate the transaction. Possible values are: <ul style="list-style-type: none"> • Click • ClickOut • Sale • Lead
RateName	String	The name of the found rate

IV. Statistics Web Service

This web service intended to obtain statistical data from the affilinet system.

<https://advertiser-webservices.affili.net/V4/WSDLFactory/Statistics.wsdl>

1. GetStatisticsPerPublisher

This method returns all transactions on the affilinet platform, grouped by publisher-ID. It has a similar functionality like the page you find on the advertiser platform under “Statistics” > “By Publisher”.

1. Parameters

GetStatisticsPerPublisherRequest

Parameter	Type	Description
DisplaySettings *	DisplaySettings	Allows you to define the paging of the results
GetStatisticsPerPublisherQuery *	GetStatisticsPerPublisherQuery	The filter settings of the statistics request

DisplaySettings

Parameter	Type	Description
CurrentPage *	Integer	Defines which result page of the size PageSize shall be returned. Counting starts at 1.
PageSize *	Integer	Defines how many records each result page shall hold. Cannot be bigger than 150.

GetStatisticsPerPublisherQuery

Parameter	Type	Description
Channel1	Integer	Limits the results to publishers, which belong to this channel 1. Input can be generated with the method "GetChannels"
Channel2	Integer	Limits the results to publishers, which belong to this channel 2. Input can be generated with the method "GetChannels"
CreativeNumber	Integer	Limits the results to transactions, which have been triggered by the creative specified with this CreativeNumber and this CreativeType. Please note that ProgramID, CreativeType and CreativeNumber form a unique identifier. If you specify the CreativeNumber, you also have to specify the CreativeType. Input can be generated with the method "GetCreativesPerType"
CreativeType	Enum	Limits the results to transactions, which have been triggered by the creative specified with this CreativeNumber and this CreativeType. Please note that ProgramID, CreativeType and CreativeNumber form a unique identifier. Possible values are: <ul style="list-style-type: none"> • Text • Banner • HTML If you specify the CreativeType, you also have to specify the CreativeNumber.
EndDate *	DateTime	Only transactions with this date or earlier will be considered for the statistics. Time span between StartDate and EndDate cannot exceed 3 months.
ProgramId *	Integer	The ID of the requesting advertiser
PublisherId	Integer	Limits the results to this single publisher
PublisherName	String	Limits the results to publishers, whose names contain this string
PublisherSegment	Integer	Only transactions triggered by publishers of this publisher segment will be considered for the statistics. Input can be

		generated with the method "GetPublisherSegments"
PublisherURL	String	Limits the results to publishers, whose URLs contain this string
RateQueryCollection	RateQueryCollection	Limits the statistics to transactions, where one of these rates was used for remuneration. Input can be generated with the method "GetRateList".
ShowDetails	Bool	Set this flag to true if you would like to receive detail information on the single leads. If set to "false", the "LeadRateDetails" property of the returned leads will be empty. Please note that setting this to "true" can lead to performance issues in large result sets. Default is "false".
StartDate *	DateTime	Only transactions with this date or later will be considered for the statistics. Time span between StartDate and EndDate cannot exceed 3 months.
ValuationType	Enum	Defines whether you want the defined StartDate and EndDate to define the day when the transactions were registered or confirmed. Possible values are: <ul style="list-style-type: none"> • Registration • Confirmation Default is Registration.
VolumeMax	Integer	Limits the statistics to publishers who have this much or less of whatever is defined in VolumeType. If VolumeMax is specified, VolumeType also needs to be filled.
VolumeMin	Integer	Limits the statistics to publishers who have this much or more of whatever is defined in VolumeType. If VolumeMin is specified, VolumeType also needs to be filled.

VolumeType	Enum	<p>Defines the statistical figure, which shall be applied to VolumeMax and/or VolumeMin. Possible values are:</p> <ul style="list-style-type: none"> • TotalViews • TotalClicks • TotalClickOuts • TotalSales • TotalLeads • TotalBonusPayments • TotalCommission • OpenCommission • ViewsRevenue • ClicksRevenue • SalesByRate • LeadsByRate • PublisherActivity * <p>*: PublisherActivity is a combination of TotalViews, TotalClicks, TotalClickOuts, TotalSales and TotalLeads. If you set VolumeType to "PublisherActivity" and specify a VolumeMin of e.g. 1, then all publishers, who did not generate a single view, click, clickOut, sale or lead in the given timespan, are NOT returned by this method – this can dramatically decrease the size of the response and is strongly recommended. Default of VolumeType is NULL.</p>
VolumeTypeRateNumber	Integer	<p>If VolumeType is set to "SalesByRate" or "LeadsByRate", this field specifies, which sale rate resp. lead rate shall be considered.</p>

RateQueryCollection

Parameter	Type	Description
RateQueryCollection	Array of RateQuery	Limits the statistics to transactions, where one of these rates was used for remuneration. Input can be generated with the method " GetRateList ".

RateQuery

Parameter	Type	Description
RateMode	Enum	Defines the type of remuneration. Possible values are: <ul style="list-style-type: none"> • Click • ClickOut • Sale • Lead Default is Click.
RateNumber	Integer	The identifier of the desired rate.

2. Return Types

GetStatisticsPerPublisherResponse

Property	Type	Description
PublisherStatisticsRecordCollection	Array of PublisherStatisticsRecord	List of all found data sets. Each record stands for one publisher.
TotalCount	Integer	The total amount of found data sets

PublisherStatisticsRecord

Property	Type	Description
Bonus	Bonus	Detailed information on the bonus payments granted to this publisher.
ClickOuts	RateModeSummary	The publisher's ClickOuts, which meet the set filter criteria.
Clicks	TrafficSummary	The publisher's Clicks, which meet the set filter criteria.
Currency	String	The currency which must be applied to all commission values.
Leads	RateModeSummary	Information on the leads of this publisher, summed up by their status ("Open", "Confirmed" or "Cancelled").

NetPriceSummary	PriceSummary	Information on the turnover which this publisher is responsible for (based on the parameter “price” of the registersale-call), summed up by the status of the transactions (“Open”, “Confirmed” or “Cancelled”).
NetworkFeeSummary	PriceSummary	Information on the network commission which is due to the transactions of this publisher, summed up by the status of the transactions (“Open”, “Confirmed” or “Cancelled”).
PublisherCommissionSummary	PriceSummary	Information on the publisher commission, summed up by the status of the transactions (“Open”, “Confirmed” or “Cancelled”).
PublisherId	Integer	The ID of the publisher represented in this data set
PublisherURL	String	The URL of the publisher represented in this data set
Sales	RateModeSummary	This publisher’s sales, which meet the set filter criteria
Views	TrafficSummary	The publisher’s Views, which meet the set filter criteria.

Bonus

Property	Type	Description
Count	TransactionSummary	The total number of bonus payments to this publisher, summed up by their status (“Open”, “Confirmed” or “Cancelled”)
NetworkFee	PriceSummary	The total monetary value of network fees calculated for bonus payments, summed up by their status (“Open”, “Confirmed” or “Cancelled”)
PublisherCommission	PriceSummary	The total monetary value of all bonus payments to this publisher, summed up by their status (“Open”, “Confirmed” or “Cancelled”)

CommissionSummary

Property	Type	Description
CancelledCommission	Double	The total commission value of all cancelled transactions of this publisher
ConfirmedCommission	Double	The total commission value of all confirmed transactions of this publisher
OpenCommission	Double	The total commission value of all open transactions of this publisher

PriceSummary

Property	Type	Description
CancelledValue	Double	The sum of net prices in cancelled transactions
ConfirmedValue	Double	The sum of net prices in confirmed transactions
OpenValue	Double	The sum of net prices in open transactions

RateDetails

Property	Type	Description
TransactionSummary	TransactionSummary	The number of transactions for the given search criteria, split up by status
Price	PriceSummary	The total NetValue (normally: shopping cart value) of all transactions matching the given search criteria, split up by status
PublisherCommission	PriceSummary	The publisher commission assigned to all transactions matching the given search criteria, split up by status
RateNumber	Integer	The rate number on which these figures are based. Details on this RateNumber can be found using the method " GetRateList ".
NetworkFee	PriceSummary	The network fee calculated for the transactions contained in TransactionSummary, split up by status

RateDetailsCollection

Property	Type	Description
RateDetailsCollection	Array of RateDetails	The Id and the amounts of open, confirmed and cancelled transactions of this lead

RateModeSummary

Property	Type	Description
RateDetailsCollection	Array of RateDetails	The amount of sales, total NetPrice, PublisherCommission and NetworkFee for each single rate number.
TotalCount	TransactionSummary	The total amount of transactions of this RateMode, summed up by their status (open, confirmed or cancelled).
TotalNetPrice	PriceSummary	The total NetPrice value of transacions of this RateMode, summed up by their status (open, confirmed or cancelled).
TotalNetworkFee	PriceSummary	The sum of NetworkFees of all transactions of this RateMode, summed up by their status (open, confirmed or cancelled).
TotalPublisherCommission	PriceSummary	The sum of PublisherCommissions, summed up by their status (open, confirmed or cancelled).

TrafficSummary

Property	Type	Description
GrossCount	Long	The amount of clicks or views before filtering was applied.
NetCount	Long	The amount of clicks or views after filtering.
NetworkFee	Double	The network commission due to these clicks or views.
PublisherCommission	Double	The publisher commission due to these clicks or views.

TransactionSummary

Property	Type	Description
CancelledCount	Long	The amount of transactions in status "Cancelled".
ConfirmedCount	Long	The amount of transactions in status "Confirmed".
OpenCount	Long	The amount of transactions in status "Open".

V. Orders Web Service

With this web service, you can get, create and edit transactions, baskets and basket items in the affilinet system.

<https://advertiser-webservices.affili.net/V4/WSDLFactory/OrderManagement.wsdl>

1. GetTransactions - DEPRECATED

This method returns all transactions on the affilinet platform, which match the given search criteria. We recommend using the method [GetOrders](#) instead, because it is suitable not only for transactions, but also for basket tracking orders.

Please note: Baskets will not get delivered as single entities, but as separate transactions, which internally represent the sale (which is why we recommend to use [GetOrder](#) instead of [GetTransactions](#)).

1. Parameters

[GetTransactionsRequest](#)

Parameter	Type	Description
CancellationReason	String	Only those transactions are returned, which contain this string in their cancellation reason.
Channel1	Integer	Limits the results to publishers, which belong to this channel 1. Input can be generated with the method "GetChannels"
Channel2	Integer	Limits the results to publishers, which belong to this channel 2. Input can be generated with the method "GetChannels"
CreativeInfo	CreativeInfo	Limits the results to transactions, which have been triggered by the creative specified with these criteria.
EndDate *	Date	Only transactions with this date or earlier will be returned. Depending on the property ValuationType, this date either considers the date when the transaction was recorded or edited. Time span between StartDate and EndDate cannot exceed 3 months.
NetPriceMax	Double	Only transactions will be returned, for which the recorded price doesn't exceed this limit.
NetPriceMin	Double	Only transactions will be returned, for which the recorded price is at least this high.
OrderId	String	The identifier of the order, as it was submitted in the registersale call.
Page *	Integer	Defines which result page of the size PageSize shall be returned. Counting starts at 1. Please note the exception if TransactionIds is

		specified (see below).
PageSize *	Integer	Defines how many records each result page shall hold. Cannot be bigger than 1000. Please note the exception if TransactionIds is specified (see below).
ProgramId *	Integer	The Id of the requesting advertiser.
PublisherCommissionMax	Double	Only transactions will be returned, through which the publisher earned this much or less.
PublisherCommissionMin	Double	Only transactions will be returned, through which the publisher earned this much or more.
PublisherIds	Array of Integer	Restricts the transactions to those, which have been generated through one of these publishers. If this is empty, the transactions of all publishers are returned. Array can hold max. 100 elements.
PublisherSegment	Integer	Only transactions triggered by publishers of this publisher segment will be returned. Input can be generated with the method “GetPublisherSegments”
RateMode	Enum	The mode of the transaction. If not defined, transactions of any mode are returned. Possible values are: <ul style="list-style-type: none"> • PayPerClickOut • PayPerSale • PayPerLead
RateNumber	Integer	The identifier of the desired rate.
StartDate *	Date	Only transactions with this date or later will be returned. Depending on the property ValuationType, this date either considers the date when the transaction was recorded or edited. Time span between StartDate and EndDate cannot exceed 3 months.
TransactionIds	Array of Integer	If specified, only transactions are returned, which have any of these IDs. The entries in “Page” and “PageSize” are ignored in this case. Array can hold up to 100 IDs.
TransactionStatus *	Enum	Restricts the returned transactions to those which have this status. Possible values of the enumeration are: <ul style="list-style-type: none"> • Open • Confirmed • Cancelled • All

ValuationType *	Enum	<p>Defines whether you want the defined StartDate and EndDate to define the day when the transactions were registered (“RegistrationDate”) or edited (“LastStatusChangeDate”).</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • RegistrationDate • LastStatusChangeDate <p>Default is RegistrationDate.</p>
------------------------	------	---

CreativeInfo

Property	Type	Description
CreativeNumber	Integer	<p>Limits the results to transactions, which have been triggered by the creative specified with this CreativeNumber and this CreativeType.</p> <p>Please note: ProgramId, CreativeType and CreativeNumber form a unique identifier.</p> <p>If you specify the CreativeNumber, you also have to specify the CreativeType, otherwise the request will fail.</p> <p>Input can be generated with the method “GetCreativesPerType”.</p>
CreativeType	Enum	<p>Limits the results to transactions, which have been triggered by this type of creative.</p> <p>Please note: ProgramId, CreativeType and CreativeNumber form a unique identifier.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • Text • Banner • HTML <p>If you specify the CreativeNumber, you also have to specify the CreativeType.</p>

2. Return Types

GetTransactionsResponse

Property	Type	Description
ProgramId	Integer	The Id of the requesting advertiser.
TotalCount	Integer	The total amount of found data sets.
TransactionCollection	Array of Transaction	List of all found data sets. Each record stands for one transaction.

Transaction

Property	Type	Description
AutoOrderManagementAction	Enum	This field tells whether or not the transaction is connected to automatic order management. And if so, to which status the transaction will be changed (Cancelled or Confirmed). Possible values are: <ul style="list-style-type: none"> • Confirm • Cancel • (null)
AutoOrderManagementActionDate	DateTime	If an AutoOrderManagementAction is configured for this transaction, this field tells when the automatic action will take place or has taken place.
CancellationReason	String	If the transaction has been cancelled or its NetPrice has been changed, this field displays the reason that has been specified for this action.
ClickDate	DateTime	The timestamp when the click cookie was placed on the customer's machine.
CreativeInfo	CreativeInfo	Information through which creative the transaction was triggered.
LastStatusChangeDate	DateTime	The date when the transaction has last been

		edited.
NetPrice	Double	This is what has been transmitted as the net value of your customer's action. For sale transactions, this is the base for the calculation of the commission. For lead transactions, NetPrice is optional.
NetworkFee	Double	The amount of money which has been remunerated to affilinet.
OrderId	String	The identifier of the order, as it was submitted by the advertiser.
PublisherCommission	Double	The amount of money which has been remunerated to the publisher.
PublisherId	Integer	The ID of the publisher to whom this transaction is assigned.
RateInfo	RateInfo	Information on the rate, which applies to this transaction.
RegistrationDate	DateTime	The date when the transaction has been added to our database.
TransactionId	Integer	The internal Id which the transaction got in the affilinet platform.
TransactionStatus	Enum	The status which the transaction currently has. Possible values are: <ul style="list-style-type: none"> • Open • Confirmed • Cancelled

CreativeInfo

Parameter	Type	Description
CreativeNumber	Integer	The Id of the creative. The combination of ProgramId, CreativeType and CreativeNumber is unique throughout the affilinet system.
CreativeType	Enum	The type of the creative. The combination of ProgramId, CreativeType and CreativeNumber is unique throughout the affilinet system. Possible values are: <ul style="list-style-type: none"> • Text • Banner • HTML

RateInfo

Parameter	Type	Description
IsTieredRate	Bool	Indicates whether or not the program is using a tiered rate system. If so, then transactions can only be confirmed by the automatic confirmation process, not manually or by a web service call.
RateDescription	String	The description of the rate. In most cases, it is the name of the default rate.
RateMode	Enum	The mode of the transaction. Possible values are: <ul style="list-style-type: none"> • PayPerClickOut • PayPerSale • PayPerLead
RateNumber	Integer	The identifier of the rate.
RateValue	Double	The value of this rate. If the rate mode is PayPerSale, this is a percentage value, otherwise it is an absolute monetary value.

2. CreateTransaction

This method generates a new transaction on the affilinet platform. That new transaction is in status “open” at first and can be changed using the method “UpdateTransactions”.

Please note: whenever a transaction is submitted to us, it is first created as a temporary transaction, then several checks are performed, e.g. if an active partnership between the advertiser and the publisher exists. Only if all checks succeed, the transaction is created. During peak times, these checks can take a couple of minutes. This is why CreateTransaction only responds with the (temporary) ProcessingId and not with the final TransactionId.

1. Parameters

CreateTransactionRequest

Parameter	Type	Description
ClickDate	DateTime	Please enter here when the click took place, which led to this transaction. If not specified, then the OrderDate is used. ClickDate has to be formatted like this: 2010-11-24T12:34:56
CreativeNumber	Integer	The Id of the creative, which was clicked on. The combination of ProgramId, CreativeType and CreativeNumber is unique throughout the affilinet system.
CreativeType	Enum	The type of the creative, which was clicked on. The combination of ProgramId, CreativeType and CreativeNumber is unique throughout the affilinet system. Possible values are: <ul style="list-style-type: none"> • Text • Banner • HTML
Currency	String	The currency which shall be applied to the monetary values specified in the call. At the time of the call, we convert this value into EUR or GBP, depending on the program’s currency. Values have to follow the ISO standard 4217 and can have 3-character-abbreviations like “CHF”, “GBP”, “USD”, “EUR”, etc., case insensitive. If a submitted value cannot get recognized, we reject the whole basket. We support all reference currencies of the ECB .
LeadMultiplier	Integer	If the commission of this lead rate shall be assigned not once but n times for this publisher, you can specify this n in the LeadMultiplier parameter. Example: a customer orders a prepaid telephone card not one, but 5 times. Creating 5 independent transactions for this process is hardly possible, because normally, each OrderId can be used only for one transaction.

		<p>Can only be used if RateMode is set to "PayPerLead", otherwise it is ignored.</p> <p>Can be any integer > 0.</p>
NetPrice	Double	The value of the action, which the customer has performed on the advertiser's platform, e.g. his shopping cart value. For sale transactions, this price is used to calculate the commissions for sale rates and is mandatory.
OrderDate	DateTime	<p>The timestamp on which the transaction was generated. Can be the current timestamp or anywhere in the last 180 days. If not specified, then the current timestamp is used.</p> <p>OrderDate has to be formatted like this: 2010-11-24T12:34:56</p>
OrderId *	String	The identifier of the transaction in the advertiser's system. Within the advertiser's system, this OrderId must be unique. Can be an alphanumeric string with up to 40 characters, case insensitive. Special characters like é, è, í, %, \$, etc. are not supported.
ProgramId *	Integer	The Id of the advertiser.
RateDescription	String	Normally orders are displayed to the publisher together with the name of the commissioned rate. If you would like the order to get a different text instead, you can submit it here in the OrderDescription. Only works for transactions, not for baskets. Can be an alphanumeric string with up to 50 characters, must be URL-encoded. Optional.
RateMode *	Enum	<p>The mode of the rate, which shall be applied to this transaction. Possible values are:</p> <ul style="list-style-type: none"> • PayPerClickOut • PayPerSale • PayPerLead
RateNumber *	Integer	The identifier of the rate, which shall be applied to this transaction.
ReferrerToken *	String	The information on the publisher, who triggered the transaction. The ReferrerToken contains the Id of the publisher and if applicable it is suffixed with "-" and the SubId, which the publisher has submitted.
SubIds	Dictionary of [String, String]	Along with each order, you can specify sub-Ids for your own purposes. Currently we support the keys pSub1 and pSub2. Both can be up to 30 characters, case insensitive. Special characters like é, è, í, %, \$, etc. are not supported. You can e.g. specify the payment method which the customer used, his time spent on your platform, a string representing his click sequence on your site, etc...

VoucherCode	String	If in the shopping process, a voucher code was used, the advertiser can submit the involved voucher code to us here. Publishers only get to see whether or not a voucher was used in the transaction, they don't see the voucher string itself. Can be an alphanumeric string with up to 25 characters, case insensitive. Special characters like é, è, í, %, \$, etc. are not supported.
--------------------	--------	---

2. Return Types

When a transaction is created by calling the method `CreateTransaction`, that transaction is saved to a temporary table and a range of checks are being performed and only if these checks are successful, it is migrated to the final orders table: e.g. whether or not a partnership with the submitted publisher exists, whether or not the rate exists, what that rate is worth for this specific publisher, ... These checks can take a couple of minutes, depending on the load on our servers.

`CreateTransactionResponse`

Property	Type	Description
ProcessingId	String	The temporary Id, which was assigned to the transaction request.
RegisterSucceeded	Bool	True if the creation of the temporary transaction was successful. Otherwise a <code>FaultException</code> will be thrown.

3. UpdateTransaction

With this method, an existing transaction can be edited: the status and the NetPrice of the transaction can be changed.

Please note: only transactions can be updated with this method. For updating BasketItems or Baskets, please use [UpdateBasketItem](#) or [UpdateBasket](#).

1. Parameters

UpdateTransactionRequest

Parameter	Type	Description
Action *	Enum	This field defines the kind of change you wish to perform: to change the status of the transaction(s) or to change the NetPrice. Possible values are: <ul style="list-style-type: none"> • SetOpen (to give it the status "Open") • SetCancelled (to give it the status "Cancelled") • SetConfirmed (to give it the status "Confirmed") • SetNewNetPrice (to change the NetPrice of the transaction)
CancellationReason	String	If a transaction is cancelled or its NetPrice is changed (if Action is "SetCancelled" or "SetNewNetPrice"), a reason for this change must be given. Must be a string of minimum length 5 non-empty characters. E.g.: <ul style="list-style-type: none"> • Failed credit check • Sale cancelled by customer • Item returned • Item out of stock • Duplicated order • Breach of campaign terms and conditions
NewNetPrice	Double	Here you can specify a new NetPrice of the transaction. Must be a Double > 0.
OrderId	String	This is the Id, which the advertiser has transmitted to affilinet for the order. Please note: either the OrderId, RateMode and RateNumber must be specified, or the TransactionId, because both are a unique identifier of the transaction in the affilinet system.
ProgramId *	Integer	The Id of the requesting advertiser
RateMode	Enum	The mode of the transaction. Possible values are: <ul style="list-style-type: none"> • PayPerClickOut • PayPerSale • PayPerLead

RateNumber	Integer	The identifier of the desired rate.
TransactionId	Integer	The unique identifier, which affilinet has assigned to the desired transaction. Please note: either the OrderId, RateMode and RateNumber must be specified, or the TransactionId, because both are a unique identifier of the transaction in the affilinet system.

2. Return Types

UpdateTransactionResponse

Property	Type	Description
Successful	Bool	True if the update request has succeeded. Otherwise a FaultException will be thrown.

4. GetOrders

This method returns all orders (that is: transactions or baskets) on the affilinet platform, which match the given search criteria.

1. Parameters

GetOrdersRequest

Parameter	Type	Description
BasketIds	Array of Integer	If specified, only baskets are returned, which have any of these Ids. The entries in "Page" and "PageSize" are ignored in this case. Array can hold up to 100 Ids. If specified, transactions are not returned, but only baskets.
CancellationReason	String	Only those orders are returned, which contain this string in their cancellation reason.
Channel1	Integer	Limits the results to publishers, which belong to this channel 1. Input can be generated with the method "GetChannels".
Channel2	Integer	Limits the results to publishers, which belong to this channel 2. Input can be generated with the method "GetChannels".
CreativeInfo	CreativeInfo	Limits the results to orders, which have been triggered by the creative specified with these criteria.
EndDate *	Date	Only orders with this date or earlier will be returned. Depending on the property ValuationType, this date either considers the date when the order was recorded or edited. Time span between StartDate and EndDate cannot exceed 3 months.
NetPriceMax	Double	Only orders will be returned, whose total net price doesn't exceed this limit.
NetPriceMin	Double	Only orders will be returned, whose total net price is at least this high.
OrderId	String	The identifier of the order, as it was submitted in the registersale call.
Page *	Integer	Defines which result page of the size PageSize shall be returned. Counting starts at 1. Please note the exception if TransactionIds is

		specified (see below).
PageSize *	Integer	Defines how many records each result page shall hold. Cannot be bigger than 1000. Please note the exception if TransactionIds is specified (see below).
ProgramId *	Integer	The Id of the requesting advertiser.
PublisherCommissionMax	Double	Only orders will be returned, through which the publisher earned this much or less.
PublisherCommissionMin	Double	Only orders will be returned, through which the publisher earned this much or more.
PublisherIds	Array of Integer	Restricts the orders to those, which have been generated through one of these publishers. If this is empty, the orders of all publishers are returned. Array can hold max. 100 elements.
PublisherSegment	Integer	Only orders triggered by publishers of this publisher segment will be returned. Input can be generated with the method “GetPublisherSegments”
RateMode	Enum	The mode of the transaction. If not defined, transactions of any mode are returned. If specified, baskets are not returned, but only transactions. Possible values are: <ul style="list-style-type: none"> • PayPerClickOut • PayPerSale • PayPerLead
RateNumber	Integer	The identifier of the desired rate. If specified, baskets are not returned, but only transactions.
StartDate *	Date	Only orders with this date or later will be returned. Depending on the property ValuationType, this date either considers the date when the order was recorded or edited. Time span between StartDate and EndDate cannot exceed 3 months.
TransactionIds	Array of Integer	If specified, only transactions are returned, which have any of these IDs. The entries in “Page” and “PageSize” are ignored in this case. Array can hold up to 100 IDs. If specified, baskets are not returned, but only transactions.
TransactionStatus *	Enum	Restricts the returned orders to those which have this status. Possible values of the enumeration are: <ul style="list-style-type: none"> • Open

		<ul style="list-style-type: none"> • Confirmed • Cancelled • All
ValuationType	Enum	<p>Defines whether you want the defined StartDate and EndDate to define the day when the orders were registered ("RegistrationDate") or edited ("LastStatusChangeDate").</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • RegistrationDate • LastStatusChangeDate <p>Default is RegistrationDate.</p>

2. Return Types

GetOrdersResponse

Property	Type	Description
OrderCollection	Array of Order	List of all found data sets. Each record stands for one transaction or basket.
ProgramId	Integer	The Id of the requesting advertiser.
TotalCount	Integer	The total amount of found data sets.

Order

Property	Type	Description
AutoOrderManagementAction	Enum	<p>This field tells whether or not the order is connected to automatic order management. And if so, to which status the transaction will be changed (Cancelled or Confirmed).</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • Confirm • Cancel • (null)
AutoOrderManagementActionDate	DateTime	If an AutoOrderManagementAction is configured for this order, this field tells when the automatic action will take place or has taken place.

BasketInfo	BasketInfo	If the returned order is a shopping basket, this object contains summary information on the current status of that shopping basket, along with its Id (which can then be used e.g. in order to obtain the single basket items with GetBasketItems).
CancellationReason	String	If the order has been cancelled or its NetPrice has been changed, this field displays the reason that has been specified for this action.
ClickDate	DateTime	The timestamp when the click cookie was placed on the customer's machine.
CookieSharingProgram	Integer	If this program shares its cookies with a different program and the other program's cookie was used when this order was created, then the Id of the program, whose cookie was used, is returned here.
CreativeInfo	CreativeInfo	Information through which creative the transaction was triggered.
LastStatusChangeDate	DateTime	The date when the transaction has last been edited.
NetPrice	Double	This is what has been transmitted as the net value of your customer's action. For sale transactions, this is the base for the calculation of the commission. For lead transactions, NetPrice is optional.
NetworkFee	Double	The amount of money which has been remunerated to affilinet.
OrderId	String	The identifier of the order, as it was submitted by the advertiser.
PreviousPublisherIds	Array of Integer	If we know of other publishers, who generated a click before the remunerated publisher, those publishers' Ids are returned here.
PublisherCommission	Double	The amount of money which has been remunerated to the publisher.
PublisherId	Integer	The ID of the publisher to whom this transaction is assigned.
RateInfo	RateInfo	Information on the rate, which applies to this transaction.

RegistrationDate	DateTime	The date when the transaction has been added to our database.
SubIds	Dictionary of [String, String]	Along with each order, advertisers can specify sub-Ids (like pSub1 or pSub2). If present for this order, they are returned here.
TransactionId	Integer	The internal Id which the transaction got in the affilinet platform.
TransactionStatus	Enum	The status which the transaction currently has. Possible values are: <ul style="list-style-type: none"> • Open • Confirmed • Cancelled
VoucherCode	String	If in the shopping process, a voucher code was used, the advertiser can submit the involved voucher code to us. If he did, it is returned here.

BasketInfo

Property	Type	Description
BasketId	Integer	The unique identifier of this basket within the affilinet system. Can be used e.g. in order to obtain the single basket items with GetBasketItems).
OpenBasketItemCount	Integer	The amount of not-cancelled basket items within this basket.
RegisteredNetPrice	Double	The total net price of all not-cancelled basket items within this basket.
TotalBasketItemCount	Integer	The total amount of basket items contained in this basket, cancelled or not-cancelled.

RateInfo

Parameter	Type	Description
IsTieredRate	Bool	Indicates whether or not the program is using a tiered rate system. If so, then its orders can only be confirmed by the automatic confirmation process, not manually or by a web service call.
RateDescription	String	The description of the rate. In most cases, it is the name of the default rate.

RateMode	Enum	The mode of the transaction. Possible values are: <ul style="list-style-type: none">• PayPerClickOut• PayPerSale• PayPerLead
RateNumber	Integer	The identifier of the rate.
RateValue	Double	The value of this rate. If the rate mode is PayPerSale, this is a percentage value, otherwise it is an absolute monetary value.

5. GetBasketItems

Through this method, single products, which are part of a shopping basket, can be obtained.

1. Parameters

GetBasketItemsRequest

Parameter	Type	Description
BasketItemsQuery *	BasketItemsQuery	Search criteria which the desired basket items ought to fulfil.
DisplaySettings *	DisplaySettings	Allows you to define the paging of the results.

DisplaySettings

Parameter	Type	Description
CurrentPage *	Integer	Defines which result page of the size PageSize shall be returned. Counting starts at 1.
PageSize *	Integer	Defines how many records each result page shall hold. Cannot be bigger than 150.

BasketItemsQuery

Parameter	Type	Description
ArticleNumber	String	If specified, then only basket items with this ArticleNumber are returned.
BasketId	Integer	The unique identifier of a single basket within the affilinet system. If specified, then only basket items are returned, which belong to this basket.
Category	String	If specified, then only basket items are returned, which were created with exactly this category information (exact matching, case sensitive).
EndDate	DateTime	If specified, only basket items will be returned, which were created with this date or earlier. Time span between StartDate and EndDate cannot exceed 3 months.
OrderId	String	The identifier of the order, as it was submitted in the registersale call.
PositionNumber	Integer	In order to access one single basket item, you can specify here the position, which the desired item has within the basket.
ProgramId *	Integer	The Id of the requesting advertiser.
RateQuery	RateQuery	If specified, then only those basket items get returned, which were remunerated with this rate.

StartDate	DateTime	If specified, only basket items will be returned, which were created with this date or later. Time span between StartDate and EndDate cannot exceed 3 months.
Status	Enum	If specified, lets you restrict to the status of the desired basket items. Possible values are: <ul style="list-style-type: none"> • cancelled • not cancelled

RateQuery

Parameter	Type	Description
RateMode *	Enum	Defines the type of remuneration. Possible values are: <ul style="list-style-type: none"> • PayPerClickOut • PayPerSale • PayPerLead Default is PayPerClickOut.
RateNumber *	Integer	The identifier of the desired rate.

2. Return Types

GetBasketItemsResponse

Property	Type	Description
BasketItemCollection	Array of BasketItem	The collection of basket items, which match the given search criteria.
TotalCount	Integer	The total amount of basket items, which match the given search criteria.

BasketItem

Property	Type	Description
ArticleNumber	String	The SKU, which this product has within the advertiser's system.
BasketId	Integer	The affilinet-wide unique Id of the basket, to which this basket item belongs.
Brand	String	The make of the bought product.

CancellationReason	Enum	The specific reason, why this item got cancelled (if it was is cancelled). Possible values are: <ul style="list-style-type: none"> • NotCancelled • FailedCreditCheck • SaleCancelledByCustomer • ItemReturned • ItemOutOfStock • DuplicatedOrder • BreachOfCampaignTermsAndConditions • Other
Category	String	The category, in which this product is kept in the advertiser's system.
IsLocked	Bool	This flag indicates whether or not this item can still be edited - that is: whether or not the basket, to which this item belongs, is still in status "open".
OriginalQuantity	Integer	This quantity was specified when this item got created - later changes to the quantity are reflected in the field Quantity.
PositionNumber	Integer	The position, which this item has within its basket.
ProductName	String	The title of the product.
Properties	Dictionary of [String, String]	Each of these 5 separate optional parameters can be used to submit additional information for this basket item. This information is also displayed to the publisher. Typical use cases is product data, which only applies to a certain subset of products - e.g. in the case of clothing, size, colour, material,... of a product can be submitted as properties.
Quantity	Integer	The current quantity of this article within this basket. Based on Quantity and SinglePrice, the net price is calculated, which is used to calculate the publisher's commission.
RateInfo	RateInfo	Information on the rate, which is used to remunerate this basket item.
SinglePrice	Double	The price, which one article of this ArticleNumber costs. Based on Quantity and SinglePrice, the net price is calculated, which is used to calculate the publisher's commission.
Status	Enum	The current status of this basket item. Possible values are: <ul style="list-style-type: none"> • NotCancelled • FailedCreditCheck • SaleCancelledByCustomer • ItemReturned

		<ul style="list-style-type: none"> • ItemOutOfStock • DuplicatedOrder • BreachOfCampaignTermsAndConditions • Other
TotalPrice	Double	The total value of this basket item, calculated as Quantity * SinglePrice.
TransactionId	Integer	Internally, this basket item belongs to a transaction. This is that transaction's Id.

6. UpdateBasket

This method allows you to change the status of a basket.

1. Parameters

UpdateBasketRequest

Parameter	Type	Description
Action *	Enum	This field defines the kind of change you wish to perform. Possible values are: <ul style="list-style-type: none"> • SetOpen (to give it the status "Open") • SetCancelled (to give it the status "Cancelled") • SetConfirmed (to give it the status "Confirmed")
BasketId	Integer	The unique identifier of a single basket within the affilinet system. In an UpdateBasketRequest, either BasketId or OrderId must be specified.
CancellationReason	String	If a basket is cancelled (that is: if Action is "SetCancelled"), a reason for this change must be given. Must be a string of minimum length 5 non-empty characters.
OrderId	String	The identifier of the transaction in the advertiser's system. Within the advertiser's system, this OrderId must be unique. Can be an alphanumeric string with up to 40 characters, case insensitive. Special characters like é, è, í, %, \$, etc. are not supported. In an UpdateBasketRequest, either BasketId or OrderId must be specified.
ProgramId *	Integer	The Id of the requesting advertiser.
PublisherId	Integer	The Id to which the basket is assigned. If multiple baskets exist with the same OrderId, the PublisherId can be used to tell them apart.

2. Return Types

UpdateBasketResponse

Property	Type	Description
Successful	Bool	True if the update request has succeeded. Otherwise a FaultException will be thrown.

7. UpdateBasketItem

This method can be used in order to cancel a single item, which is part of a basket (or later un-cancel it again) by choosing the suitable CancellationReason, or to change the quantity of a single basket item.

Please note: Basket items can only be updated when the basket, to which they are assigned, is in status “open”. Otherwise you will receive an error message.

1. Parameters

UpdateBasketItemRequest

Parameter	Type	Description
ArticleNumber	String	The SKU this item has within the advertiser’s system. Can be an alphanumeric string of up to 80 characters. In an UpdateBasketItemRequest, either BasketId and PositionNumber or OrderId and ArticleNumber must be specified.
BasketId	Integer	The affilinet-wide unique identifier of the basket, to which the affected basket item belongs. In an UpdateBasketItemRequest, either BasketId and PositionNumber or OrderId and ArticleNumber must be specified.
CancellationReason	Enum	If an item is cancelled, a reason for this change must be given. Possible values are: <ul style="list-style-type: none"> • NotCancelled • FailedCreditCheck • SaleCancelledByCustomer • ItemReturned • ItemOutOfStock • DuplicatedOrder • BreachOfCampaignTermsAndConditions • Other
OrderId	String	The identifier of the transaction in the advertiser’s system. Within the advertiser’s system, this OrderId must be unique. Can be an alphanumeric string with up to 40 characters, case insensitive. Special characters like é, è, í, %, \$, etc. are not supported. In an UpdateBasketItemRequest, either BasketId and PositionNumber or OrderId and ArticleNumber must be specified.
PositionNumber	Integer	The position of the affected item within the basket (can be obtained through GetBasketItems). In an UpdateBasketItemRequest, either BasketId and PositionNumber or OrderId and ArticleNumber must be specified.
ProgramId *	Integer	The Id of the requesting advertiser.

Quantity	Integer	The count of bought articles with this ArticleNumber. We multiply Quantity with SinglePrice to calculate the NetPrice, based on which the publisher's commission is calculated.
-----------------	---------	---

2. Return Types

UpdateTransactionsResponse

Property	Type	Description
Successful	Bool	True if the update request has succeeded. Otherwise a FaultException will be thrown.

8. CreateBasket

As an alternative to the integration of a tracking pixel or something equivalent on the order confirmation page of the advertiser, new baskets can also be created with this method.

Please note: whenever a new order is submitted to us, it is first created as a temporary order, then several checks are performed, e.g. if an active partnership between the advertiser and the publisher exists. Only if all checks succeed, the order is created. During peak times, these checks can take a couple of minutes.

1. Parameters

CreateBasketRequest

Parameter	Type	Description
BasketItemCollection *	Array of RegisterBasketItemData	The detailed information on the single basket items involved in this shopping process.
ClickDate	DateTime	Please enter here when the click took place, which led to this order. If not specified, then the OrderDate is used. ClickDate has to be formatted like this: 2010-11-24T12:34:56
CreativeNumber	Integer	The Id of the creative, which was clicked on. The combination of ProgramId, CreativeType and CreativeNumber is unique throughout the affilinet system.
CreativeType	Enum	The type of the creative, which was clicked on. The combination of ProgramId, CreativeType and CreativeNumber is unique throughout the affilinet system. Possible values are: <ul style="list-style-type: none"> • Text • Banner • HTML
Currency	String	The currency which shall be applied to the monetary values specified in the call. At the time of the call, we convert this value into EUR or GBP, depending on the program's currency. Values have to follow the ISO standard 4217 and can have 3-character-abbreviations like "CHF", "GBP", "USD", "EUR", etc., case insensitive. If a submitted value cannot get recognized, we reject the whole basket. We support all reference currencies of the EZB .

OrderDate	DateTime	The timestamp on which the order was generated. Can be the current timestamp or anywhere in the last 180 days. If not specified, then the current timestamp is used. OrderDate has to be formatted like this: 2010-11-24T12:34:56
OrderId	String	The identifier of the transaction in the advertiser's system. Within the advertiser's system, this OrderId must be unique. Can be an alphanumeric string with up to 40 characters, case insensitive. Special characters like é, è, í, %, \$, etc. are not supported.
ProgramId *	Integer	The Id of the advertiser.
ReferrerToken	String	The information on the publisher, who triggered the transaction. The ReferrerToken contains the Id of the publisher and if applicable it is suffixed with "-" and the SubId, which the publisher has submitted.
SubIds	Dictionary of [String, String]	Along with each order, you can specify sub-Ids for your own purposes. Currently we support the keys pSub1 and pSub2. Both can be up to 30 characters, case insensitive. Special characters like é, è, í, %, \$, etc. are not supported. You can e.g. specify the payment method which the customer used, his time spent on your platform, a string representing his click sequence on your site, etc...
VoucherCode	String	If in the shopping process, a voucher code was used, the advertiser can submit the involved voucher code to us here. Publishers only get to see whether or not a voucher was used in the transaction, they don't see the voucher string itself. Can be an alphanumeric string with up to 25 characters, case insensitive. Special characters like é, è, í, %, \$, etc. are not supported.

RegisterBasketItemData

Parameter	Type	Description
ArticleNumber *	String	The SKU this item has within the advertiser's system. Can be an alphanumerical string of up to 80 characters. Exceeding characters will be cut off.
Brand	String	The make of the bought product.

Category	String	The category of the advertiser's system, in which the bought product is located, including category path, if available. E.g. "Electronics > Portable devices > Mp3 players". Can be an alphanumerical string of up to 255 characters. Exceeding characters will be cut off.
ProductName *	String	The title of the product, e.g. "Apple iPod Nano 8 GB". Can be an alphanumerical string of up to 255 characters. Exceeding characters will be cut off.
Properties	Dictionary of [String, String]	Additional information on the bought product. Up to 5 additional fields can be specified. This information is also displayed to the publisher. Typical use cases is product data, which only applies to a certain subset of products - e.g. in the case of clothing, you could submit here the size, colour, material,... of the product. Each property can contain up to 255 characters.
Quantity *	Integer	The count of bought articles with this ArticleNumber. We multiply Quantity with SinglePrice to calculate the NetPrice, based on which the publisher's commission is calculated. If e.g. somebody bought the same article in the same order process 5 times, then this parameter Quantity must be set to 5. Can be any integer > 0.
RateIdentifier	RateIdentifier	If you wish to explicitly specify, which rate shall be used when remunerating the sale of this product, you can specify it here. If you don't, then this program's rate mapping settings will be applied.
SinglePrice *	Double	The price, which one item of this ArticleNumber costs. We multiply Quantity with SinglePrice to calculate the NetPrice, based on which the publisher's commission is calculated. Cannot be negative, but can be 0.

RateIdentifier

Parameter	Type	Description
RateMode	Enum	The mode of the rate, which shall be applied when remunerating the sale of this product. Possible values are: <ul style="list-style-type: none"> • PayPerClickOut • PayPerSale • PayPerLead
RateNumber	Integer	The identifier of the rate, which shall be applied when remunerating the sale of this product.

2. Return Types

CreateBasketResponse

Property	Type	Description
----------	------	-------------

BasketId Integer The Id of the basket, which got created.

Please note: several logical checks are performed after the basket got created, e.g. whether or not a partnership with the referring publisher exists. So although a CreateBasketRequest was syntactically OK, a basket might still not get created (e.g. if no partnership existed between the submitted program and publisher, or a non-existing rate was specified, etc.).

VI. Sample Code

1. Java

This java example comes with 2 files for user authentication (UsernameTokenHandler.java and UserPasswordInfo.java) and a Main.java. It requires webservicess-rt.jar and webservicess-tools.jar, both of which are part of METRO (<https://metro.dev.java.net/>).

1. UsernameTokenHandler.java

```
package advertiserstatisticsclient;

import javax.xml.namespace.QName;
import javax.xml.ws.handler.MessageContext;
import javax.xml.ws.handler.soap.*;
import java.util.*;

public class UsernameTokenHandler
implements SOAPHandler<SOAPMessageContext>
{
    public Set<QName> getHeaders()
    {
        return null;
    }

    public boolean handleMessage(SOAPMessageContext context)
    {
        Boolean isOutbound =
        (Boolean) context.get(context.MESSAGE_OUTBOUND_PROPERTY);
        if(isOutbound.booleanValue())
        {
            context.put(com.sun.xml.wss.XWSSConstants.USERNAME_PROPERTY,
            UserPasswordInfo.getUsername());
            context.put(com.sun.xml.wss.XWSSConstants.PASSWORD_PROPERTY,
            UserPasswordInfo.getPassword());
        }
        return true;
    }

    public boolean handleFault(SOAPMessageContext context)
    {
        return true;
    }

    public void close(MessageContext context)
    {
    }
}
```

2. UserPasswordInfo.java

```
package advertiserstatisticsclient;

public class UserPasswordInfo {
    private static String username;
    private static String password;

    public static String getPassword()
    {
        return password;
    }

    public static void setPassword(String password)
    {
        UserPasswordInfo.password = password;
    }

    public static String getUsername()
    {
        return username;
    }

    public static void setUsername(String username)
    {
        UserPasswordInfo.username = username;
    }
}
```

3. Main.java

```
package advertiserstatisticsclient;

import com.microsoft.schemas._2003_10.serialization.arrays.ArrayOfKeyValueOfstringstring;
import esbfa.framework.affilinet.advertiserservices.ordermanagement.OrderManagementContract;
import esbfa.framework.affilinet.advertiserservices.ordermanagement.OrderManagementContractGetTransactionsOrderManagementFaultFaultFaultMessage;
import esbfa.framework.affilinet.advertiserservices.ordermanagement.OrderManagementService;
import esbfa.framework.affilinet.advertiserservices.ordermanagement.messages.GetTransactionsRequest;
import esbfa.framework.affilinet.advertiserservices.ordermanagement.messages.GetTransactionsResponse;
import esbfa.framework.affilinet.advertiserservices.ordermanagement.types.TransactionStatus;
import esbfa.framework.affilinet.advertiserservices.ordermanagement.types.ValuationType;
import java.util.List;
```

```
import javax.xml.bind.JAXBElement;
import javax.xml.datatype.DatatypeFactory;
import javax.xml.datatype.XMLGregorianCalendar;
import javax.xml.namespace.QName;
import javax.xml.ws.BindingProvider;
import javax.xml.ws.handler.Handler;

/**
 * wsdlLocation: https://advertiser-webservices.affili.net/V3/WSDLFactory/OrderManagement.wsdl
 */

public class Main {

    private static String nameOfMessages =
        "http://affilinet.framework.esbfa/advertiserservices/ordermanagement/messages";

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {

        try {
            getTransactions();
        }
        catch(Exception e) {
            System.out.println(e.getMessage());
        }
    }

    /**
     * Samplecode to call the GetTransactions servicemethod
     */
    public static void getTransactions() {
        try {
            /* set your credentials here:
             * The credentials are set, using the custom UsernameTokenHandler,
             * which is used to add your credentials to the messagecontext of your requestmessage.
             * Here, the wssecurity standard is used.
             */
            UserPasswordInfo.setUsername("abc");
            UserPasswordInfo.setPassword("secret");
            System.setProperty("com.sun.xml.ws.transport.http.client.HttpTransportPipe.dump", "true");

            /*
             * The OrderManagementService defines the operations you use to get or
             * manipulate your transactions.
             */
        }
    }
}
```

```
OrderManagementService service = new OrderManagementService();

/*
 * The port is defined through the OrderManagementContract,
 * which defines the interface to the serviceoperations.
 */
OrderManagementContract port = service.getPort(OrderManagementContract.class);

/*
 * This code sets the context of your request
 * and adds the custom UsernameTokenHandler (see credential-section above)
 * so the messagehandler-chain is manipulated to use wssecurity with your cedentials.
 */
BindingProvider prov = (BindingProvider) port;
List<Handler> handlerChain = prov.getBinding().getHandlerChain();
handlerChain.add(new UsernameTokenHandler());
prov.getBinding().setHandlerChain(handlerChain);
prov.getRequestContext().put(prov.ENDPOINT_ADDRESS_PROPERTY, "https://advertiser-
webservices.affili.net/advertiserservice.svc");

/*
 * This part creates your request-parameters.
 */
XMLGregorianCalendar startDate =
    DatatypeFactory.newInstance().newXMLGregorianCalendar(2010, 7, 1, 0, 0, 0, 0);
XMLGregorianCalendar endDate =
    DatatypeFactory.newInstance().newXMLGregorianCalendar(2010, 7, 31, 0, 0, 0, 0);
QName transactionStatusQName =
    new QName(nameOfMessages, TransactionStatus.class.getSimpleName());
JAXBElement<TransactionStatus> transactionStatus =
    new JAXBElement<TransactionStatus>(
        transactionStatusQName,
        TransactionStatus.class,
        TransactionStatus.ALL);
QName valuationTypeQName =
    new QName(nameOfMessages, ValuationType.class.getSimpleName());
JAXBElement<ValuationType> valuationType =
    new JAXBElement<ValuationType>(
        valuationTypeQName,
        ValuationType.class,
        ValuationType.REGISTRATION_DATE);

/*
 * This part sets your request-parameters.
 */
GetTransactionsRequest request = new GetTransactionsRequest();
request.setStartDate(startDate);
request.setEndDate(endDate);
```

```

request.setPage(1);
request.setPageSize(2);
request.setProgramId(123456);
request.setTransactionStatus(transactionStatus);
request.setValuationType(valuationType);

/*
 * The service is called, and the response will be returned.
 */
GetTransactionsResponse response =
    port.getTransactions(request);

/*
 * Process the response as you like.
 * This sample only prints the number of total results
 * and the number of currently in the response contained results.
 */
System.out.println("Total transactions matching the request-parameters: " +
    response.getTotalCount());
System.out.println("Number of transactions in this response: " +
    response.getTransactionCollection().getValue().getTransaction().size());
}
catch (OrderManagementContractGetTransactionsOrderManagementFaultFaultFaultMessage ex) {
    HandleException(ex);
}
catch (Exception ex2) {
    System.out.println(ex2.getMessage());
    if (ex2.getStackTrace() != null) {
        System.out.println("STACKTRACE: ");
        for (StackTraceElement se : ex2.getStackTrace()) {
            System.out.println(se.getClassName() + " - " + se.getMethodName() + " - Line(" + se.getLineNumber() + ")");
        }
    }
}
}

private static void HandleException(
    OrderManagementContractGetTransactionsOrderManagementFaultFaultFaultMessage ex) {
    System.out.println("Add your handling-code here: ");
    System.out.println(ex.getMessage());

    ArrayOfKeyValueOfstringstring arr =
        ex.getFaultInfo().getErrorMessages().getValue();
    List<ArrayOfKeyValueOfstringstring.KeyValueOfstringstring> kvpList =
        arr.getKeyValueOfstringstring();
    for (ArrayOfKeyValueOfstringstring.KeyValueOfstringstring kvp : kvpList)
    {

```

```
        System.out.println(kvp.getKey() + " - " + kvp.getValue());  
    }  
}
```

2. C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.ServiceModel;
using System.Xml;
using System.Xml.Serialization;

namespace AdvWS_Orders_Live
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                GetTransactions();
            }
            catch (Exception e)
            {
                Console.WriteLine(e.ToString());
            }
            finally
            {
                Console.WriteLine("Press Enter to EXIT ...");
                Console.ReadLine();
            }
        }

        static void GetTransactions()
        {
            // add a Service Reference to your project by using the following wsdl-location:
            // https://advertiser-webservices.affili.net/V3/WSDLFactory/OrderManagement.wsdl
            // AdvWSOrders <= the name of the ServiceReference we used in this example code

            AdvWSOrders.GetTransactionsRequest request =
                new AdvWSOrders.GetTransactionsRequest();
            request.StartDate = new DateTime(2010, 7, 1);
            request.EndDate = new DateTime(2010, 7, 31);
            request.Page = 1;
            request.PageSize = 10;
            request.ProgramId = 123456789;
        }
    }
}
```

```
request.ValuationType = AdvWSOrders.ValuationType.RegistrationDate;
request.TransactionStatus = AdvWSOrders.TransactionStatus.All;

// The ChannelFactory is used to create a channel for sending messages to the service endpoint.
// It has nothing to do with channels an advertiser can use to group programs.
// The name used in the Constructor is the name of the endpoint in your app.config which
// is used to create a connection to the service.
ChannelFactory<AdvWSOrders.OrderManagementContract> factory =
    new ChannelFactory<AdvWSOrders.OrderManagementContract>(
        "BasicHttpBinding_OrderManagementContract");
factory.Credentials.UserName.UserName = "abc"; // enter your advertiser-webservice-username here
factory.Credentials.UserName.Password = "secret"; // enter the corresponding password here

// created the channel to the specified endpoint-address
AdvWSOrders.OrderManagementContract serviceInstance = factory.CreateChannel();

try
{
    AdvWSOrders.GetTransactionsResponse response =
        serviceInstance.GetTransactions(request);

    // add your business intelligence here to interpret the response
}
finally
{
    try
    {
        ((IClientChannel)serviceInstance).Close();
        factory.Close();
    }
    catch (Exception e)
    {
        Console.WriteLine("Problem closing the service-channel: \n" + e.ToString());
    }
}
}
```

3. PHP

```
<?php

define ("WSDL_SERVICE", "https://advertiser-webservices.affili.net/V3/WSDLFactory/Toolbox.wsdl");

// Namespace
$ns = 'http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd';
//Security Utility
$wsu = 'http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd';
// web service user name, defined in the account section in the affilinet portal
$username = 'MyUsername';
// password of that web service user
$password = 'MyPassword';

// set time to GMT
$created = date
    ("Y-m-d\TH:i:s", mktime (date("H") +5, date("i"), date("s"), date("m"), date("d"), date("Y")))."Z";

$token = new stdClass;
$token->Username = new SOAPVar($username, XSD_STRING, null, null, null, $ns);
$token->Password = new SOAPVar($password, XSD_STRING, null, null, null, $ns);
$token->Created = new SOAPVar($created, XSD_STRING, null, null, null, $wsu);

$wsec = new stdClass;
$wsec->UsernameToken = new SoapVar($token, SOAP_ENC_OBJECT, null, null, null, $ns);

$headers = new SOAPHeader($ns, 'Security', $wsec, true);

$SOAP_REQUEST = new SoapClient(WSDL_SERVICE);
$SOAP_REQUEST->__setSoapHeaders($headers);
$response = $SOAP_REQUEST->GetPublisherSegments(array(
    'ProgramId' => 000 // MyProgramID
));

print_r($response);

?>
```