

BusinessMonitor[®] Connector

Technical documentation

Versie 2020.jul.08

Contents

1	Connection information/examples	3
	Endpoint URL	3
	Authentication	3
	C# example	3
	PHP example	4
	Other languages	4
2	BusinessMonitor® service methods	5
	GetAuthKey (username, password)	5
	GetAuthApiKey (api key)	5
	CheckKey	5
	HelloWorldTest	6
	CreateEmailList	6
	CreateEmailMessage	6
	AddRecordToEmailList	6
	CreateEIDForUser	7
	EditEmailList	7
	CreateUniqueResultsUrl	7
	GetEID	7
	GetStatus	8
	GetSurveys	8
	IsCompleted	8
	ValidateMxEmailAddress	8
	ValidateMobilePhoneNumber	9
	ScheduleEmailMessage	9
	GetAggregateRating	10
	GetAggregateRatingFilter	11
	GetReviews	12
	GetReviewsFilter	12
	GetReviewsFilterM	13
	GetReviewsFilter2Filters	13
	GetReviewsFilter2FiltersM	13
	GetIndividualReports	14
3	BusinessMonitor® record objects	15
	EmailRecord	15
	EmailMessageRecord	15
	AggregateRating	15
	Review	16
	SurveyRecord	16

IndividualReport.....	16
4 Usages for the BusinessMonitor® connector.....	16
Generating an individual link.....	17
Scheduling an imported e-mail list.....	19
Presenting reviews on your website, social channels, comparison sites.....	20

1 Connection information/examples

This chapter will give some examples on how to connect with the BusinessMonitor® connector.

Endpoint URL

The endpoint URL that has to be used for adding the service reference is:

<https://s.businessmonitor.nl/bmservice.svc>

<https://s.businessmonitor.nl/bmservice.svc?wsdl>

<https://s.businessmonitor.nl/bmservice.asmx>

<https://s.businessmonitor.nl/bmservice.asmx?wsdl>

Always use `https://` for the endpoints.

Authentication

The API supports logins by Username/Password combination or by using an API key, both of these generate a temporary key to use with the method calls. See these 2 chapters for information on authenticating `GetAuthKey`, `GetAuthApiKey`. Where a method needs a "key" to be used this would be the key returned by `GetAuthKey` or `GetAuthApiKey`, and not the API key used for logging in.

C# example

When you want to connect to the BusinessMonitor® connector via C# you can import the service via the visual studio 'Add Service Reference' function when you right click on the project you wish to add it to. This will make it possible to call the methods in code as follows:

```
BmServiceClient client = new BmServiceClient();
client.HelloWorldTest(key);
```

For example a call to get the Aggregate Rating for a given item Id:

```
BmWebservice.BmServiceSoap bms = new BmServiceSoapClient();

GetAuthApiKeyRequest gkr = new GetAuthApiKeyRequest();
gkr.Body = new GetAuthApiKeyRequestBody("APIKEY", "127.0.0.1");
var key = bms.GetAuthApiKey(gkr).Body.GetAuthApiKeyResult;

GetAggregateRatingRequest gr = new GetAggregateRatingRequest();
gr.Body = new GetAggregateRatingRequestBody(123456, key);
var x = bms.GetAggregateRating(gr);

form1.Controls.Add(new Literal() {Text =
x.Body.GetAggregateRatingResult.RatingValue.ToString()});
```

PHP example

If you want to connect to the BusinessMonitor® connector using PHP you can use the following example:

```
$client = NEW SoapClient('https://s.businessmonitor.nl/bmservice.asmx?WSDL' ,
Array('trace'=>1));

$client->GetAuthKey(Array('username' => 'username', 'password' => 'password', 'ip'
=> 'ip address'));
```

For example a call to get the Aggregate Rating for a given item Id:

```
<?php
# Customer details
$ip= '127.0.0.1';
$itemId = 'YOUR_ITEM_ID';
$apiKey = 'YOUR_API_KEY';

# Setup the soap connection
$client = NEW
SoapClient('https://s.businessmonitor.nl/bmservice.asmx?WSDL',Array('trace'=>1));

# Generate a temporary key (this key is valid for 24 hours)
$result = $client->GetAuthApiKey(Array('apiKey' => $apiKey, 'ip' => $ip));
$keyResultset = get_object_vars($result);
#var_dump($keyResultset);
$key = $keyResultset['GetAuthApiKeyResult'];

# Get the aggregateRating object
$ratingResult = $client->GetAggregateRating(Array('itemId'=>$itemId,'key'=>$key));
$ratingResultset = get_object_vars($ratingResult);
#var_dump($ratingResultset);
$ratingValue = round($ratingResultset['GetAggregateRatingResult']->RatingValue,1);
#var_dump($ratingValue);
$ratingMin = $ratingResultset['GetAggregateRatingResult']->MinimumValue;
$ratingMax = $ratingResultset['GetAggregateRatingResult']->MaximumValue;
$ratingCount = $ratingResultset['GetAggregateRatingResult']->ReviewCount;
?>
```

Other languages

For more examples on how to connect with an asmx webservice you can visit:
<http://geekswithblogs.net/THines01/archive/2010/03/07/callws.aspx>

2 BusinessMonitor® service methods

The following methods are currently present in the BusinessMonitor® connector.

GetAuthKey (username, password)

This method will generate an authentication key that is needed to call the other methods. In order to generate the key, you will need to provide your BusinessMonitor username and password including the ip address.

Please keep in mind that an auth key is only valid for 24 hours.

- Input : username - The username of the api user
password - The password of the api user
ip - The ip address
- Output : string with the key

If this method returns an empty string the password is incorrect or the user is locked out from the system.

GetAuthApiKey (api key)

This method will generate an authentication key that is needed to call the other methods. In order to generate the key, you will need to provide your BusinessMonitor ApiKey including the ip address.

Gets the authentication key based on the api key.

- Input : string apiKey - The apiKey
ip - The ip address
- Output : string with the key when successful login, else returns empty string

CheckKey

This method can be used to check if a key is still active in the database. In order to check if the key is still active, you can send in your auth api key. If it is still active you will receive a boolean with the value true.

- Input : string key The auth api key
- Output : boolean true/false True if they auth key is still active

HelloWorldTest

This method is created for testing purposes. The method returns a string with your current user ID.

- Input : string key
- Output : string helloworldText
- Comments : The text that will be returned is:
 - 'Hello user. Your User ID is: + userID'

Pseudo code flow for this method would be:

```
var tempKey = GetAuthApiKey(api key)
echo HelloWorldTest(tempKey)
```

CreateEmailList

This method will create an email list for the user. This method has 2 parameters that are required. You must provide the emailListName and the key to create a new email list. The other parameters are optional. When the email list is successfully created, the new emailListID is returned.

- Input : string emailListName, string description, string CustomData1, string CustomData2, string CustomData3, string key
- Output : int emailListID
- Comments : -

CreateEmailMessage

This method can be used to create an email message. The email message record you will have to provide is discussed in the next chapter. This method will return an emailMessageID that can be used in the 'CreateEIDForUser' method.

- Input : EmailMessageRecord¹ emailMsgRecord, string key
- Output : int emailMessageID

AddRecordToEmailList

This method can be used to add a record to an email list. The email record you will have to provide is discussed in the next chapter. This method will return an emailAddressID that can be used in the 'CreateEIDForUser' method.

- Input : EmailRecord² emailRecord, string key
- Output : int emailAddressID (the id of the record as saved in our database)

Error	Explanation
ERROR: The key that was provided does not exist.	The provided key has timed out or is not supplied, request a new one from GetAuthApiKey

¹ The specifications of the EmailMessageRecord object can be found in the next chapter

² The specifications of the EmailRecord object can be found in the next chapter

ERROR: The provided email list ID is not linked to your account.	You tried to add records to an e-mail list the user is not connected to as admin
ERROR: The email address needs to be entered in in order to add the record to the email list.	EmailRecord is not supplied

CreateEIDForUser

This method will generate a unique ID that can be used within a URL to send a user to the corresponding survey.

- Input : int surveyId, int emailAddressId, emailMessageId, string key
- Output : string UniqueID
- Comments : The id can be used in an url as follows: t.asp?EID=<UniqueID>

EditEmailList

This method will allow you to edit an email list by providing the email list ID of the email list that needs to be updated along with the new values. *Note:* you will reset the other fields to 'empty' if you do not provide any values. This means that you need to provide all the fields that you want to set to a specific value.

- Input : int emailListID, string emailListName, string description, string CustomData1, string CustomData2, string CustomData3, string key
- Output : bool EditSucceeded

CreateUniqueResultsUrl

This method can be used to create an unique url so the user can view the results for the given e-mail list, please keep in mind that this is a 'locked' filter so the user cannot view the report if there are no answers yet.

- Input : int surveyId – the survey
int emailListId – the email list that should be locked
string key – the auth api key
- Output : The url to use for the (total) report
- Comments : Example output:
 - ResultsOverview.asp?SID=kjdghfkdshfdkjshfkjdshfkjdsd

GetEID

This method allows the user to determine what the most recent EID value is for a given E-mail address id.

- Input : String key – The auth api key
Int emailAddressId – The email address id
- Output : String with the 'newest' EID for the specific email address id.
- Comments : If there is no message sent, an error will be returned:
 - ERROR: Email address id not found in messages.

GetStatus

This method can be used to determine the status of the 'unique' link.

Gets the status of the given unique Id [no response, responded, declined]. If no response the user did not open the link, if responded the user opened the link and answered at least one question, declined the user declined the invitation and as such did not want to participate.

- Input : String key – The auth api key
String eid – The unique Id for the respondent
String emailAddressId – The email address id
- Output : The current status for the given unique eid
- Comments : The output is one of the following:
 - [No response, Responded, Declined]

GetSurveys

This method will get a list of surveys for the current user or another user that is managed by the current user.

- Input : int subUserId – 0 for the current user or an user id for the list of the subuser
String key – The auth api key
- Output : A list of SurveyRecords available in the system
- Comments : SurveyRecord is documented in the next chapter

IsCompleted

This method can be used to determine if the user completed the survey based on the 'unique' link. Keep in mind that it will only return true/false if the user did respond to the invitation, so GetStatus has to return 'Responded' for this method to return a Boolean value.

Determines whether there is a completed response for the specified eid.

- Input : String key – The auth api key
String eid – The unique id for the respondent (eid)
String emailAddressId – The email address id
- Output : Boolean true if the given eid has a completed response else false

ValidateMxEmailAddress

Method to validate one e-mail address to determine if the tld of the e-mail address has an mx server (if not we can be sure the e-mail will bounce)

- Input : String key – The auth api key
String emailAddress – The e-mail address to validate
- Output : Boolean false when the e-mail address is invalid for sure, in any other case true will be returned

ValidateMobilePhoneNumber

Method to validate one string to determine if it could be a phone number

- Input : String key – The auth api key
String mobileNumber – The phone number to validate
- Output : Boolean false when the phone number is invalid for sure, in any other case true will be returned

ScheduleEmailMessage

Schedules an email message for the given email list to the specified survey. Our sending jobs run hourly, an e-mail message will only be included when the scheduled time has passed. If the job runs on 08:15 and the mail is scheduled for 08:30 it will be included with the mailings of 09:15.

The given Survey needs to have an e-mail template configured and the e-mail list must be added to the system.

- Input : key – The auth api key
emailListId – The email list identifier
surveyId - The survey identifier
invite - The invite DateTime
reminder - The reminder DateTime, empty if no reminder
- Output : DateTime of the invite e-mail

The format for the value in DateTime is: YYYY-MM-DDTHH:MM:SS

Error messages:

ERROR: User is not logged in.

ERROR: Access denied, not admin of the email list.

ERROR: Access denied, not admin of the survey.

ERROR: Survey has no e-mail template configured.

ERROR: Reminder must be after the invite.

GetAggregateRating

Gets the aggregate rating for the given ItemId.

Returns an object with the average value, number of ratings, minimum value, and the maximum value.

Can be used inside websites e.g. using JSON-LD or with some styling and micro-data to generate the rating for use with google ads

<http://schema.org/AggregateRating>

- Input : String key – The auth api key
String itemId – The item identifier
- Output : AggregateRating object

Supported item types:

3 Choice one answer

17 Matrix-1-Answer-Per-Row (answer values need to be configured) – keep in mind that all sub items are totalized

21 Matrix-Number keep in mind that all sub items are totalized

If the error ITEM_TYPE_NOT_YET_SUPPORTED is returned either the item is not supported yet or the user is not authorized to request the rating for the item.

To include the ratings to the website the following snippet could be configured (and used on the index page), marked values should vary on **the organization** and the values **returned from the API**.

```
<script type="application/ld+json">
{
"@context": "http://schema.org",
"@type": "Organization",
"aggregateRating": {
"@type": "AggregateRating",
"worstRating": "1",
"bestRating": "5",
"ratingValue": "4.7",
"reviewCount": "49"
},
"name": "BusinessMonitor",
"telephone": "+31102802800",
"url": "https://businessmonitor.nl"
}
</script>
```

GetAggregateRatingFilter

Gets the aggregate rating for the given ItemId.

Returns an object with the average value, number of ratings, minimum value, and the maximum value.

Can be used inside websites e.g. using JSON-LD or with some styling and micro-data to generate the rating for use with google ads

<http://schema.org/AggregateRating>

- Input : String key – The auth api key
String itemId – The item identifier
filterField – Custom data field where the filter should be applied
filterValue – value to filter on (if left empty no filter is applied)
- Output : AggregateRating object

Supported item types:

3 Choice one answer

17 Matrix-1-Answer-Per-Row (answer values need to be configured) – keep in mind that all sub items are totalized

21 Matrix-Number keep in mind that all sub items are totalized – keep in mind that all sub items are totalized

If the error ITEM_TYPE_NOT_YET_SUPPORTED is returned either the item is not supported yet or the user is not authorized to request the rating for the item.

To include the ratings to the website the following snippet could be configured (and used on the index page), marked values should vary on **the organization** and the values **returned from the API**.

```
<script type="application/ld+json">
{
"@context": "http://schema.org",
"@type": "Organization",
"aggregateRating": {
"@type": "AggregateRating",
"worstRating": "1",
"bestRating": "5",
"ratingValue": "4.7",
"reviewCount": "49"
},
"name": "BusinessMonitor",
"telephone": "+31102802800",
"url": "https://businessmonitor.nl"
}
</script>
```

GetReviews

Gets the 50 newest reviews for the given items.

Can be used inside websites e.g. JSON-ld or with some styling and micro-data to generate the rating for use with google ads

<http://schema.org/Review>

- Input : key – The auth api key - used to check if the feed is accessible
itemGrade – The item id of the item that contains the review grade
itemText - The item id of the item that contains the review text
itemAgree - The item the user answers to agree that their review can be used on the public web
answerAgree - The answer id of the "agree" to be used answer
fieldWho - The custom data field that contains the name
fieldProduct - The custom data field that contains the product/course etc. name
- Output : Review[] List with reviews

Error messages:

ERROR: User is not logged in.

ERROR: fieldProduct is not a valid field.

ERROR: fieldWho is not a valid field.

ERROR: multiple surveys returned.

ERROR: survey not found.

GetReviewsFilter

Gets the 50 newest reviews for the given items filtered on the supplied filter.

Can be used inside websites e.g. JSON-ld or with some styling and micro-data to generate the rating for use with google ads

<http://schema.org/Review>

- Input : key – The auth api key - used to check if the feed is accessible
itemGrade – The item id of the item that contains the review grade
itemText - The item id of the item that contains the review text
itemAgree - The item the user answers to agree that their review can be used on the public web
answerAgree - The answer id of the "agree" to be used answer

fieldWho - The custom data field that contains the name

fieldProduct - The custom data field that contains the product/course etc. name

filterField - Custom data field where the filter should be applied

filterValue - value to filter on (if left empty no filter is applied)

● Output : Review[] List with reviews

Error messages:

ERROR: User is not logged in.

ERROR: fieldProduct is not a valid field.

ERROR: fieldWho is not a valid field.

ERROR: multiple surveys returned.

ERROR: survey not found.

ERROR: filterField is not a valid field.

GetReviewsFilterM

Same method/parameters as GetReviewsFilter, but supports an array of int values for answerAgree. Makes it possible that there are multiple answers that agree to forwarding. E.g. "yes, anonymous & "yes, with name"

Also supports multiple string values for the filterValue

GetReviewsFilter2Filters

Same as GetReviewsFilter but supports 2 extra parameters, to make it possible to filter on a second custom data field.

Parameter: filterField2

The custom data field to filter, e.g.: CUSTOM_DATA_13

Parameter: filterValue2

The value to filter in the custom data field e.g. NL

GetReviewsFilter2FiltersM

Same method/parameters as GetReviewsFilter2Filters, but supports an array of int values for answerAgree. Makes it possible that there are multiple answers that agree to forwarding. E.g. "yes, anonymous & "yes, with name"

Also supports multiple string values for the filterValue / filterValue 2

GetIndividualReports

Used to get a list of individual responses for the specified e-mail address. The list is returned in the order "newest – oldest"

Input variable	Contents
emailAddress	e-mail address to get the individual links for. E.g. info@businessmonitor.nl
key	The auth api key - used to check if the results are accessible

Returns a list with IndividualReport objects

3 BusinessMonitor® record objects

The BusinessMonitor® connector uses different types of record objects that contain the fields that are used to create the email lists and to add the email addresses. These objects are specified below.

EmailRecord

The e-mail Record object is used for adding a new email address to a specific email list. An email record contains the following fields:

Type	FieldName	Required
string	email	Yes
int	emailListID	Yes
string	Firstname	No
string	Lastname	No
string	Custom_Data_01 t/m Custom_Data_99	No

EmailMessageRecord

The emailMessageRecord object is used for creating a new email message in the database. This message needs to be created if you want to generate the survey url for a specific user. An email message record contains the following fields:

Type	FieldName	Required	Note
int	EmailListID	Yes	
int	SurveyID	Yes	
string	EmailSubject	No	
string	EmailBody	No	
string	EmailCcAddresses	No	
string	EmailBccAddresses	No	
string	EmailFromAddress	No	
bool	HtmlYN	No	Not used, obsolete

AggregateRating

The AggregateRating object is used to return the rating for a specific item id, it will return the min, max, average and the count.

Type	Fieldname	Required
Double	RatingValue	Yes
Int	MinimumValue	Yes
Int	MaximumValue	Yes
Int	ReviewCount	Yes

Review

Name	Description
maximumValue	The maximum value
minimumValue	The minimum value
reviewBody	The review body
reviewProduct	The product/service the review is about
reviewRating	The review rating
reviewWho	The person writing the review

SurveyRecord

(This type is not completely documented here)

Name	Description
SurveyID	The ID of the survey
Title	The title of the survey e.g. "Questionnaire A" (the Title is shown to respondents)
SubTitle	The subtitle of the survey e.g. "after x months" (Subtitle is not shown to respondents)
Status	The current status of the survey, this can be Closed/Design/Open. Only open surveys accept answers from respondents (if it's not expired)
CreatedDate	Date when the survey was <i>created</i>
LaunchedDate	Date the survey when the survey was <i>initially launched</i>
StartDate	Date the survey <i>opens</i> for respondents
EndDate	Date the survey <i>closes</i> for respondents
ClosedDate	Date the survey was <i>closed</i>

IndividualReport

The individual report object is used for returning links to individual responses on the BusinessMonitor platform.:

Type	FieldName	Description
int	responseId	The responseId for the given record
int	surveyId	The surveyId the response is a part of
DateTime	responseStartDate	Date time the response is entered into the system
string	displayUrl	Full uri of the individual report for the given response

4 Usages for the BusinessMonitor® connector

This chapter contains example usages of the BusinessMonitor® connector.

Generating an individual link

When you want to generate an individual URL to a certain survey with the BusinessMonitor connector, you will have to take the following steps:

Create a survey using the BusinessMonitor® survey tool.

Create a new email list using the 'CreateEmailList' method provided by the BusinessMonitor® connector and retrieve the email list id.

Add email addresses to the email list using the email list id to generate the email address id using the 'AddRecordToEmailList' method.

Create an email message using the email list id and the survey id to generate the corresponding email message id using the 'CreateEmailMessage' method.

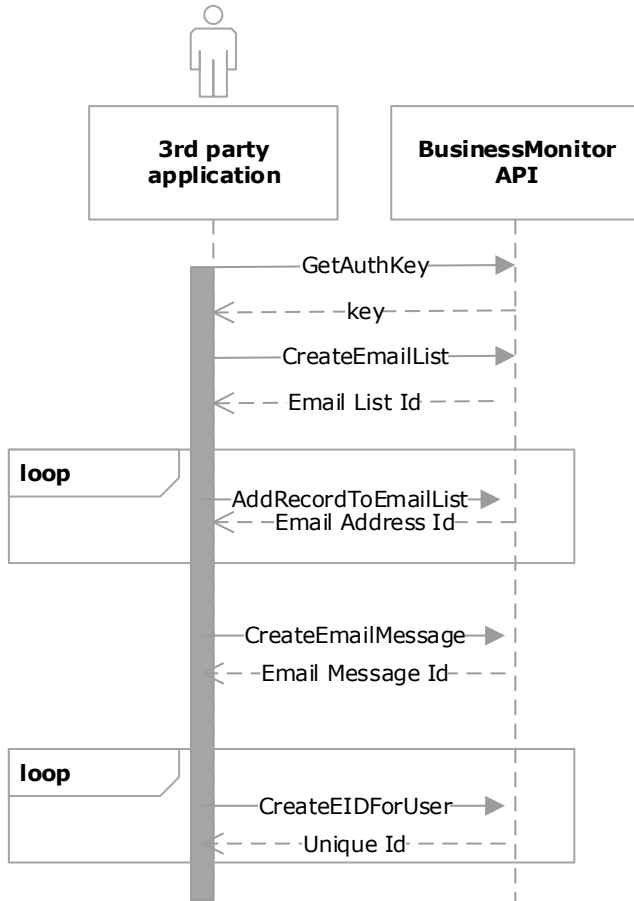
Generate the UniqueID that can be used to create the survey url by using the email list id, the email message id and the email address id of the specified user. The UniqueId can be generated using the 'CreateEIDForUser' method.

Generate the url in the following format: `http://survey.yourwebsite.com/t.asp?EID=<UniqueID>`
survey.yourwebsite.com should be a cname to s.businessmonitor.nl.
Or redirect the user to `https://bm1.nl/t.asp?EID=<UniqueID>`

Be sure to cache the used values such as EmailAddressId and the UniqueID in your own database for use with other methods.

Step 4 can be before step 3 as it's only required once.

Unieke link per deelnemer aanmaken



Scheduling an imported e-mail list

When you want to automate importing e-mail lists and scheduling the list for e-mailings the API can be used.

Pre conditions:

The survey is created using the BusinessMonitor® survey tool

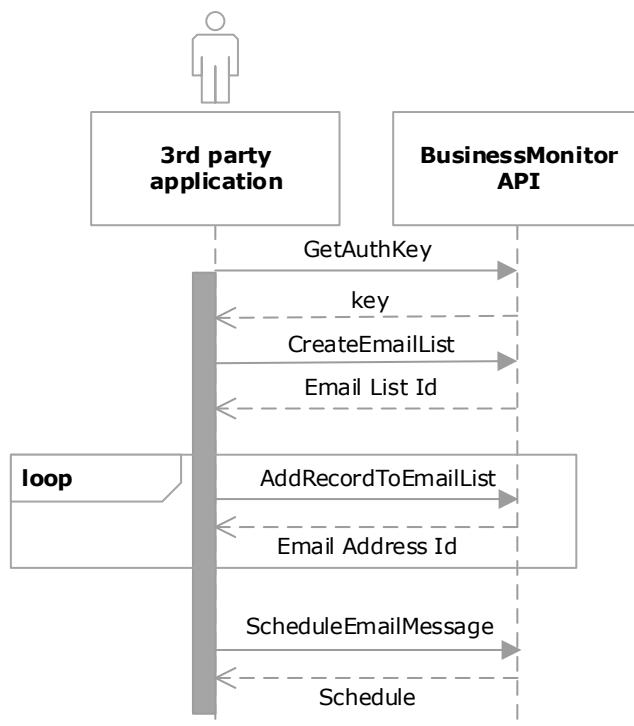
An e-mail template is configured using the BusinessMonitor® survey tool

Create a new email list using the 'CreateEmailList' method provided by the BusinessMonitor® connector and retrieve the email list id.

Add email addresses to the email list using the email list id to generate the email address ids using the 'AddRecordToEmailList' method.

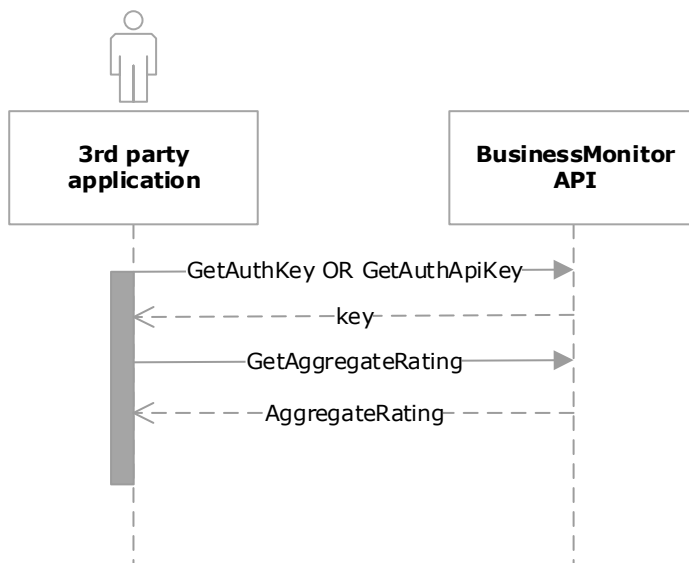
Schedule the e-mail list to the survey using the 'ScheduleEmailMessage' method.

Lijst importeren en inplannen



Presenting reviews on your website, social channels, comparison sites

AggregateRating ophalen



Some sample widget ideas for usage with the Aggregate Rating, and the review feed.

Opleiderscore

★ ★ ★ ★ ☆

8.1

Beoordeling door BusinessMonitor



8.1

Evaluaties van BusinessMonitor



Evaluaties van **BusinessMonitor**

26 klanten beoordeelden ons:

 **9.6 / 10**