Documentation eKomi | API

Professional Customer Feedback Management

Content

1. Introduction

eKomi develops and operates intelligent feedback systems. Using social commerce technology, eKomi collects customer feedback, product reviews, comments and recommendations. These are used to generate positive viral effects on websites, in social media and in search engines which increases customer trust and therefore sales.

eKomi's feedback system can be accessed on-line using an API based on web standards. This document describes how to use eKomi's API to connect to the eKomi feedback system.

1.1 Implementation Overview

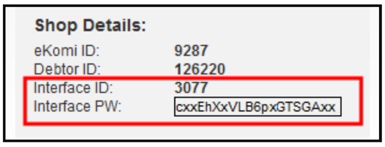
The first step to connecting to eKomi's feedback service is the creation of feedback links. A feedback link is a unique URL where a customer can leave feedback for your service as well as for any products purchased. The API methods needed for link creation are **putOrder** and **putProduct**. Next, the feedback link must be transmitted to you customer. The **getSettings** method is available to retrieve the email settings saved in your customer area.

After feedback has been received and reviewed by eKomi's Customer Feedback Management team, there are several methods available to find out more about your feedback score and who has left a review. The **getSnapshot** method returns information about your account and feedback score as well as the last 10 provider reviews received. **getRated** returns a list of order IDs for which a review has been left.

Finally, there are ways to access your reviews and display them on your website. When you are implemented with eKomi you will have access to an eKomi widget for displaying your reviews and review score on your website. In addition, the **get_feedback** script is available to download all of your provider feedback. The **get_productfeedback** script is available to download all product feedback.

1.1 Getting Started

To connect to eKomi's API, you will need an interface ID and password. This information can be found in your eKomi customer area on the right side:



Shop Details:

eKomi ID:	9287
Debtor ID:	126220
Interface ID:	3077
Interface PW:	cxxEhXxVLB6pxGTSGAxx

The core features of eKomi's API are accessible through a web-standard SOAP interface. The SOAP-endpoint of the eKomi APIs is:

http://api.ekomi.de/v2/wsdl

Each of these SOAP methods requires your API ID and API key separated by a pipe and a script version. For custom implementations, please begin your version with the text "cust-", for example: "cust-1.0.0".

2. Getting implemented

The most important part of an eKomi implementation is generating feedback links. Whether you are only collecting reviews about the shopping experience itself (**provider reviews**) or also about the individual products or services purchased (**product reviews**) only a single feedback link will be generated which is linked to a unique transaction number (order ID).

2.1. Provider Reviews

The **putOrder** method is used to generate a unique feedback link from a given order ID. This method expects the following parameters (required parameters are in grey, optional parameters in blue):

Parameter	Description	Example
auth	Interface ID and interface password separated by a pipe (|)	665|asdf123ghjk465789
version	Script version. For custom implementations, begin the version with "cust-"	cust-1.0.0
order_id	Unique transaction or order ID	79234
product_ids	Only for product reviews. A comma separated list of product IDs.	4168,357,189

The response is a serialized PHP array with the following elements:

Parameter	Type	Description
link	string	Feedback link including hash to a feedback form for the given order_id
hash	string	The customer hash for the given order_id
known_since	int	Time the order_id was first sent as a UNIX timestamp or 0 if this is the first transfer
done	int	1
done_at	int	UNIX timestamp of successful completion

2.2. Product Reviews

To collect product reviews, each product must first be registered in the eKomi review system using the **putProduct** method. The **putProduct** method expects the following parameters (required parameters are in grey, optional in blue):

Parameter	Description	Example
auth	Interface ID and interface password separated by a pipe (|)	665|asdf123ghjk465789
version	Script version. For custom implementations, begin the version with "cust-"	cust-1.0.0
product_id	Product ID	4168
product_name	Product name	Blue Keyring
product_other	Generally an empty string. See below for details.	

The response is a serialized PHP array with the following elements:

Parameter	Type	Description
done	int	1
done_at	int	UNIX timestamp of successful completion

Note 1: Only one product may be entered at a time. Please see below for example implementations.

Note 2: You may enter a product as many times as you want. If the same product ID is entered again with a different description, the new description will replace the old one.

2.2.1. Additional product information

Along with the name and ID, it is possible to send additional product information to eKomi. Depending on the agreement you have with eKomi, this additional information can be used to show an image of the product on the feedback page or it might be passed on to Google for better Google product integration. Please check with your eKomi project manager for the details.

This additional information is sent as a PHP serialized array with the following optional keys:

Key	Description
image_url	Must be an https server. Maximum resolution 150 x 150px
brand_name	Product brand name
product_ids	Array with the keys: mpc, upc, ean, isbn, gbase
links	Links to the product page and images. See example
categories	Product category. See example

Example product other array :

```
$products_other = array(
   'image_url'=>'https://www.example.com/images/thisProduct_150x150.jpg', // HTTPS, max. 150x150px
  'brand_name'=>, // Brand name
  'product_ids'=>array(
     'mpn'=>, // Manufacturer's Part Number
     'upc'=>, // Universal Product Code
     'ean'=>, // European Article Number
     'isbn'=>, // International Standard Book Number
     'gbase'=> // Google BaseID
  ), // product IDs
  'links'=>array(
     array('rel'=>'canonical', 'type'=>'text/html', 'href'=>'http://www.example.com/view_product-12'), // Link to
the product
     array('rel'=>'related', 'type'=>'image/(gif|jpg)',
'href'=>'http://www.example.com/images/thisProduct_fullsize.gif') // additional Images, could occur more
often
  ), // ProductLinks
  'categories'=>array(
     1=>array('name'=>'Consumer Electronic', 'id'=>'123'),
     2=>array('name'=>'Digital Cameras', 'id'=>'456')
  ), // categories  key>level, range:1-4 - id=[optional]
  'research'=>array(
    'add' =>array('research_id=>'1234', 'arrange'=>'1'),
              array('research_id=>'1235', 'arrange'=>'2')
  ) // Only if your package includes additional product research questions
);
```

2.3. Email settings

The most common way to transmit a feedback link to your customer is by email. eKomi's customer area offers a convenient central location where email settings can be maintained. You can access the email settings made in the customer area using the **getSettings** method.

getSettings expects the following parameters :

Parameter	Description	Example
auth	Interface ID and interface password separated by a pipe (|)	665|asdf123ghjk465789
version	Script version. For custom implementations, begin the version with "cust-"	cust-1.0.0

The response is a serialized PHP array with the following elements:

Parameter	Type	Description
mail_subject	string	Email subject
mail_html	string	Email text as HTML
mail_plain	string	Email text as plain text
mail_delay	string	Email delay
mail_from_name	string	Sender name
mail_from_email	string	Sender email
ekomi_certificate	string	Link to eKomi certificate page
ekomi_certificate_seo	string	Link to eKomi SEO certificate page
done	int	Was the action successfully completed? Possible values: 0|1

3. Getting Feedback Statistics

Two API methods are available to find out more about your feedback score and who has left a review or not. **getSnapshot** provides information about your account and a list of the last 10 reviews received. **getRated** returns a list of who has left feedback.

3.1. getSnapshot Method

The **getSnapshot** method is provided for customers who wish to create their own custom widgets, but the information it returns can be used for many different applications, for example to create a ticker of the most recent reviews received. If you wish to create a custom widget, please ask your eKomi project manager about the details.

getSnapshot expects the following parameters:

Parameter	Description	Example
auth	Interface ID and interface password separated by a pipe (|)	665|asdf123ghjk465789
version	Script version. For custom implementations, begin the version with "cust-"	cust-1.0.0

The response is a serialized PHP array with two sub-arrays, "info" and "feedback". These arrays have the following elements:

Parameter	Type	Description
info	array	Account information
info[account_name]	string	Account name
info[ekomi_certificate]	string	URL of certificate page
info[ekomi_certificate_id]	string	ID of certificate page
info[ekomi_certificate_seo]	string	URL of SEO certificate page
info[fb_count]	int	Number of reviews received

info[fb_avg]	float	Official average rating
info[fb_avg_detail]	float	Exact average rating
info[fb_avg_room]	float	Average room rating (Hotel only)
info[fb_avg_service]	float	Average service rating (Hotel only)
info[fb_avg_catering]	float	Average catering rating (Hotel only)
info[fb_avg_ambience]	float	Average ambience rating (Hotel only)
feedbacks	array	Last 10 reviews
feedbacks[0-9]	array	Review details
feedbacks[0-9][transaction_id]	string	order_id given in putOrder
feedbacks[0-9][rating]	int	Star rating (1-5)
feedbacks[0-9][rating_room]	int	Star rating room (1- 5) (Hotel only)
feedbacks[0-9][rating_service]	int	Star rating service (1-5) (Hotel only)
feedbacks[0-9][rating_catering]	int	Star rating catering (1-5) (Hotel only)
feedbacks[0-9][rating_ambience]	int	Star rating ambience (1-5) (Hotel only)
feedbacks[0-9][message]	string	Review text
feedbacks[0-9][comment]	string	Your comment
feedbacks[0-9][delivered]	datetime	1
done_at	int	UNIX timestamp of successful completion

3.2. getRated Method

The **getRated** method returns a list of the order IDs for which a review has been left within a given period. With this information you can see who has left a review or not which allows you to, for example, send a reminder email to customers who haven't left a review yet, or send a thank you email to customers who have already left a review.

Note: This method returns a list of all order IDs for which a review has been received, even if that review hasn't been approved by eKomi's Customer Feedback Management team yet. If you only want to know which customers have left a review which has already been published, please use the **get_feedback.php** or **get_productfeedback.php** calls described below.

The **getRated** method expects the following parameters:

Parameter	Description	Example
auth	Interface ID and interface password separated by a pipe (|)	665|asdf123ghjk465789
version	Script version. Can be up to 30 chars long. For custom implementations, begin the version with "cust-"	cust-1.0.0
days	How many days should be returned? Value must be between 0 and 90. The default value is 8.	20

The response is a serialized PHP array containing the order IDs associated with the last reviews received for the number of days specified.

4. Accessing Your Feedback

In addition to **getSnapshot** mentioned above, several API calls are available to give you access to reviews collected through the eKomi feedback system. As with the SOAP methods, each of these API calls will require your unique interface ID and interface password which you can find in your eKomi customer area. These scripts are called from the following address:

http://api.ekomi.de/

4.1. Accessing Provider Reviews

Provider reviews can be accessed using the **get_feedback** API call.

Note: This script is not designed for live integration into web pages. We recommend calling this script once a day and caching the results on your local server to ensure the most reliable page load times.

The **get_feedback** call expects the following parameters (required parameters are in grey, optional in blue):

Parameter	Description	Example
interface_id	The interface ID from your customer area	665
interface_pw	The interface password from your customer area	asdf123ghjk465789
type	Currently the only option is "csv". This returns an RFC 4180 formatted, comma-separated, csv (delimiter=, enclosure=").	csv
filter	Possible values are "all", "positive" and "critical". A positive review has 4 or 5 stars and a critical review has 3 or less. Default is "all".	all
charset	Which character set should be used? Possible values are "iso" and "utf-8". Default is utf-8 without BOM.	utf-8
caching	How long should the generated file be cached? Possible values are "none", "short", "middle", "long" and "verylong" which correspond to 0s, 120s, 300s, 600s and 3600s. Please only use this for testing. get_feedback is not designed for live integration.	short
range	Limit reviews returned to a certain period of time. Allowable values are 1, 3 or 6 months (1m, 3m, 6m), 1 year (1y) or all. Default is all.	1m

Example:

http://api.ekomi.de/get_feedback.php?interface_id=665&interface_pw=asdf123ghjk465789&version=cust-1.0.0&type=csv

get_feedback returns an RFC 4180 formatted, comma-separated, csv (delimiter=, enclosure=") with five columns: UNIXTIME,order_id,rating,feedback,comment.

Example csv file:

```
1217255100,1020,5,"Super!",
1217255160,1021,4,"Everything was fine",
1217255160,1023,5,"I will recommend you to my friends!"
1217255220,1024,5,"As described, fast delivery",
```

4.2. Accessing Product Reviews

Reviews about individual products can be accessed using the **get_productfeedback** API call.

Note: This script is not designed for live integration into web pages. We recommend calling this script once a day and caching the results on your local server to ensure the most reliable page load times.

The **get_productfeedback** call expects the following parameters (required parameters are in grey, optional in blue):

Parameter	Description	Example
interface_id	The interface ID from your customer area	665
interface_pw	The interface password from your customer area	asdf123ghjk465789
type	Currently the only option is "csv". This returns an RFC 4180 formatted, comma-separated, csv (delimiter=, enclosure=").	csv
product	Restrict the results to reviews for a specific product. Default is "all".	all
charset	Which character set should be used? Possible values are "iso" and "utf-8". Default is utf-8 without BOM.	utf-8
caching	How long should the generated file be cached? Possible values are "none", "short", "middle", "long" and "verylong" which correspond to 0s, 120s, 300s, 600s and 3600s. Please only use this for testing. get_productfeedback is not designed for live integration.	short
range	Limit reviews returned to a period. Allowable values are 1, 3 or 6 months (1m, 3m, 6m), 1 year (1y) or all. Default is all.	1m

Example:

http://api.ekomi.de/get_productfeedback.php?interface_id=665&interface_pw=asdf123ghjk465789&version=cust-1.0.0&type=csv

get_productfeedback returns an RFC 4180 formatted, comma-separated, csv (delimiter=, enclosure=") with five columns: UNIXTIME,order_id,product_id,rating,feedback.

Example csv file:

```
1296817039,1020,1002224,5,Great product
1297083289,1020,10011403,3,I didn't like the color
1297083805,1021,10011403,5,Great quality, but about half a size bigger than expected
1297083805,1021,1002240,5,Would buy again.
```

4.3. Accessing Campaigns

In addition to provider and product feedback, it is also possible to create custom surveys which will be included in the review process. The results of these surveys can be accessed with the **get_research** API call.

Note: This script is not designed for live integration into web pages. We recommend calling this script once a day and caching the results on your local server to ensure the most reliable page load times.

The **get_research** call expects the following parameters (required parameters are in green, optional in blue):

Parameter	Description	Example
interface_id	The interface ID from your customer area	665
interface_pw	The interface password from your customer area	asdf123ghjk465789
type	Currently the only option is "csv". This returns an RFC 4180 formatted, comma-separated, csv (delimiter=, enclosure=").	csv
content	Options are "results", "questions", "answers", "campaigns" or "campaign_questions". Default is "results".	csv
charset	Which character set should be used? Possible values are "iso" and "utf-8". Default is utf-8 without BOM.	utf-8
range	Limit results to a certain period. Allowable values are 1, 3 or 6 months (1m, 3m, 6m), 1 year (1y) or all. Default is all.	1m

Example:

http://api.ekomi.de/get_research.php?interface_id=665&interface_pw=asdf123ghjk465789&version=cust-1.0.0&type=csv

get_research returns an RFC 4180 formatted, comma-separated, csv (delimiter=, enclosure="). The number of csv columns depend on the value given for content:

Parameter	Columns
results	UNIX_TIMESTAMP, order_id, research_id, answer_id, input
questions	research_id, research_type, question
answers	research_id, answer_id, text
campaigns	campaign_id, campaign_title
campaign_questions	campaign_id, research_id

4.4. Accessing Product Market Research

Along with the usual stars and comment, it is possible to ask customers additional questions about each product purchased. Please note that this feature is not available to all customers. Please speak with your eKomi project manager. The results of these surveys can be accessed with the **get_productresearch** API call.

Note: This script is not designed for live integration into web pages. We recommend calling this script once a day and caching the results on your local server to ensure the most reliable page load times.

The **get_productresearch** call expects the following parameters (required parameters are in grey, optional in blue):

Parameter	Description	Example
interface_id	The interface ID from your customer area	665
interface_pw	The interface password from your customer area	asdf123ghjk465789
type	Currently the only option is "csv". This returns an RFC 4180 formatted, comma-separated, csv (delimiter=, enclosure=").	csv
content	Options are "results", "questions" and "answers". Default is "results".	results

charset	Which character set should be used? Possible values are "iso" and "utf-8". Default is utf-8 without BOM.	utf-8
range	Limit reviews returned to a period. Allowable values are 1, 3 or 6 months (1m, 3m, 6m), 1 year (1y) or all. Default is all.	1m

Example:

http://api.ekomi.de/get_productresearch.php?interface_id=665&interface_pw=asdf123ghjk465789 &type=csv

get_productresearch returns an RFC 4180 formatted, comma-separated, csv (delimiter=, enclosure="). The number of csv columns depend on the value given for content:

Parameter	Columns
results	UNIX_TIMESTAMP, order_id, product_id, research_id, answer_id, input
questions	research_id, research_type, question
answers	research_id, answer_id, text

5. Implementation Examples

There are many ways to integrate the eKomi API into you website. The following examples are meant to illustrate sample use cases, but should not be understood as hard and fast rules for how the API must be implemented. Your personal implementation will depend on several factors including your shop system, the programming language you use and your internal work flow.

5.1. Example 1: Simple Implementation

For this example, we will assume order information is stored on a central table in your shop system and that there is a column which is updated when an order has been shipped to a customer. This will be true for many shop systems.

We will create one table and one script and modify a script which is probably already running on your system now.

✓ **Set up an eKomi table**

First we will set up a table for our eKomi feedback links. This table will contain entries for **order ID** and **eKomi feedback link**.

✓ **Set up a cron job**

Next, we will create a cron job to run a script once a day. When the script runs, it will open the orders table and look for orders with the status "shipped". If it finds any orders which haven't been added yet, one by one it will call putOrders and generate an eKomi feedback link for that order, then order ID and feedback link will both be added to the table. It would also be possible to directly trigger the creation of a feedback link as soon as the order status is changed.

✓ **Sending the feedback link to the customer**

You might already send a follow up email to your customers. If so, it would be possible to modify the script which generates this email to include the eKomi feedback link into this email. If not, you will probably want to create a separate email to request feedback which is sent after some delay to ensure that the customer has received his order and had a chance to evaluate it. In this case we will add additional fields to the eKomi table for time read and time sent. We will create a script which runs once a day and looks for orders in the table for which the time sent hasn't been sent and the time read is older than the delay we choose. We recommend using the **getSettings** API method to access the email template and settings you made in your eKomi customer area. There will be three placeholders, [vorname], [nachname] and [ekomilink]. We will replaced these placeholders with the customer's first name and last name (or however we wish to address the customer) and the eKomi feedback link for that order. After this is done, the script will send the email through our email server.

5.2. Example 2: Product reviews

With example 1 we only collected reviews about the shop itself, but not about the products purchased. To also collect product reviews, we need to register each product with the eKomi review system using the **putProduct** API method before creating the feedback link.

- ✓ **Register products**

This can either be done dynamically, registering the products of each customer as they are shipped, or we can register all the products at once and only add new products as they are entered into the system, it's up to you. If the product description changes, simply register it again using **putProduct**. The new description will replace the old one.

- ✓ **Include product IDs when creating the feedback link**

After we have registered the products, we only need to include their IDs when we create a feedback link with **putOrder** to request product feedback from the customer. It is possible to include the order IDs of products which have not been registered with the eKomi feedback system, but the customer will not be asked to leave feedback for these products. There is not a separate feedback link, both provider and product feedback is left with the same feedback link.

- ✓ **Access the product reviews and store them locally**

After product reviews have been collected, we need to access them and store them locally. First we will download the reviews using the **get_productfeedback** API call. We can either temporarily store the csv returned, or simply store it in RAM before parsing it and storing the results in a database table. We recommend using the "range" parameter to limit the search to the last month to reduce download time and server load.

- ✓ **Integrate product reviews into the product pages**

The final step is to actually integrate reviews for each product into its product page for the best SEO effect. This can be done dynamically by checking the local product review table for any entries found each time the product page is loaded. There are many ways to integrate these reviews into the page and countless examples can be found on-line.

5.3. Example 3: iFrame integration

Often customers don't need any additional time to evaluate their order. In these cases it may make more sense to ask them for feedback immediately after purchasing. One possible way to do this is with an iFrame solution.

- ✓ **Create the link**

When the product or service is purchased, we immediately generate a feedback link using **putOrder**. If product reviews should also be collected, then these products need to be registered before creating the feedback link.

- ✓ **Create the iFrame**

After the link has been created, this link can be immediately called in an iFrame, for example on the last page of the order process.