Trusted Shops API

Application Programming Interface

Draft Version 0.5

Version 0.5 international

TRUSTED SHOPS®
The safe way to web shopping

Table of Contents

1. Introduction to Trusted Shops

The Trusted Shops trustmark on a merchant website increases online customers' confidence in the online retailer's reliability and generates increasing willingness to make a purchase. The included Trusted Shops Buyer Protection provides customers a guaranteed purchase. With Trusted Shops Customer Reviews, online buyers have the opportunity to rate an online shop using a rating and review form. In addition, the online shop can display their ratings publicly. As a neutral entity, Trusted Shops audits online shops all across Europe for their adherence to consumer legal regulations. Over 15,000 shops already carry the Trusted Shops trustmark.

Trusted Shops supports online retailers to fulfil the more than 100 quality criteria for creditworthiness, data protection security and the obligation to provide legal information. At the beginning of the audit process (certification), the online retailers are supplied with extensive practical resources such as sample texts, step-by-step guides and checklists. The quality criteria are based on European and national law as well as recommendations from consumer protection organisations.

Trusted Shops membership is currently offered for Austria, France, Italy, Germany, the Netherlands Poland, Spain, Switzerland and United Kingdom as well as an international version for all European Union countries.

2. The Trusted Shops API

2.1 Business Purpose

The intention is the Trusted Shops API will in future provide access to all Trusted Shops products. By enabling access to granular business functionalities, we want to offer our customers the possibility to integrate Trusted Shops seamlessly into their processes and systems.

2.2 Made for easy integration

By offering a RESTful web service, we want to reduce our customers' integration efforts. Relying on the robust and easy-to-use HTTP standard, RESTful web services can be implemented regardless of the platform in use. It is best practice and widely accepted amongst developers.

Keeping the structure simple, we will offer entities of our business object model as resources and a set of operations to interact with those:

Example for Trusted Shops RESTful service

```
GET https://api.trustedshops.com/rest/restricted/v1/shops/TSID/reviews.json
```

We use self-descriptive URLs and rely on HTTP standards. As you might guess, the above operation will lead you to a restricted area. Authentication is needed here. Afterwards you will have access to your shops latest reviews in JSON format. It is as easy as that. You just have to replace "TSID" with your Trusted Shops ID. We do - of course - also offer an XML output.

3. The Trusted Shops Customer Reviews API

3.1 Status of the API

We will start by offering access to your shops' latest reviews. This is a pilot and we are happy to learn from your feedback. Step by step, we will introduce new resources and operations. This also means, that the API is subject to change. There is a clear versioning within the URL and we will endeavour to keep the changes backwards compatible. However, we will keep all users of the API updated on the latest developments.

3.2 Tips for Integration

The structure of the API's response will stay the same, however we may add new attributes and entities. Please do not rely on indexes or the current sort order of the entities within the response. As we do not change the tree structure and hierarchy, we will likely be changing the sort order of the response. It is best to address values by their corresponding attribute names.

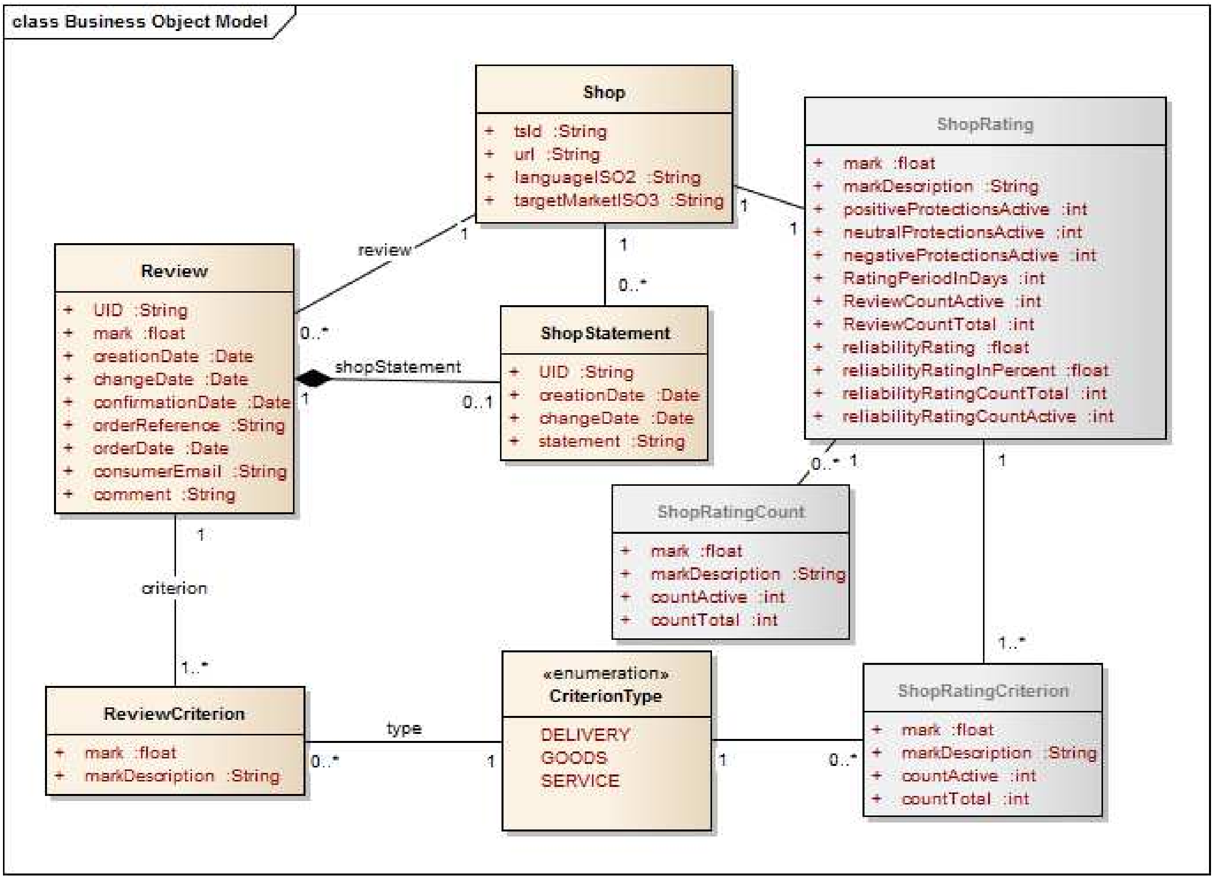
Please also notice:

- For future compatibility all attributes named "mark" are delivered as float and should be treated as float, despite the fact that these are integers at the moment.
- All attributes with data type Date are ISO8601 compliant and therefore in the format `yyyy-MM-dd'T'HH:mm:ssZ`.
- Dates supplied with the parameters startDate and endDate are in the format `yyyy-MM-dd` (disposing the time related data).

4. Information Model

4.1 Information Model Diagram

The following diagram shows the underlying information model of the Customer Reviews API. The entities in grey are currently not accessible via the API, but will follow shortly. For more information, please see 4.2 Entity and Attribute description.



4.2 Entity and Attribute description

Shop	An online shop, distinguished by language and targetMarket
tsId	The unique identifier of Trusted Shops for a shop
url	The URL of the shop
languageISO2	The language of the shop in ISO 639 format with two characters, all lowercase.
targetMarketISO3	The main target market of the shop in ISO 3166 with three characters, all uppercase.
Review	With a review, a consumer states his opinion towards a shop by rating several criteria and giving feedback with a written comment.
UID	Unique identifier for this review. This is not included in the first version of the API! Will be added March 2014.

mark	The average calculated from the single marks of all review criteria, which belong to this review.
markDescription	The mark in words, possible values are: “EXCELLENT”, “GOOD”, “FAIR”, “POOR”, “VERY_POOR”.
creationDate	Date of the review creation
changeDate	Date the review was changed on (if it was changed by the consumer at all)
confirmationDate	Date on which the review was confirmed by the consumer via double opt-in.
orderReference	Order number from the shop
orderDate	The date of the order. Only available for shops with targetMarket FRA.
consumerEmail	The consumer’s mail address
comment	Free text field of 400 characters, where the consumer can state his opinion regarding the shops’ performance
ReviewCritierion	A review consists of several ReviewCriterion which represent a CriterionType (Goods, Delivery, Service). Consumers can rate these critieria on a scale from 1-5. If a CriterionType is not rated at all, it will be left out from the response.
mark	Every ReviewCriterion type can be rated. Consumers can rate each criterion with a mark on a scale from 1-5.
markDescription	The mark in words, possible values are: “EXCELLENT”, “GOOD”, “FAIR”, “POOR”, “VERY_POOR”.
CriterionType	This is the concrete type a ReviewCriterion can have like Goods, Delivery or Service.
name	This is the concrete type a ReviewCriterion can have like Goods, Delivery or Service.
ShopStatement	A shops’ comment on a consumer's review.
statement	The actual text of the comment
UID	Unique identifier of the ShopStatement. This is not included in the first version of the API.
creationDate	The creation date of the statement. This is not included in the first version of the API.
changeDate	The change date of the ShopStatement. This is not included in the first version of the API.

The following table describes the entities which will be included in the second iteration:

ShopRating	Aggregates all reviews collected by a shop. Enriched with data from the Trusted Shops guarantee (protections), it conveys a picture of the shops performance within the last rating period.
mark	Average of all review marks of a shop.
markDescription	The mark in words, possible values are: “EXCELLENT”, “GOOD”, “FAIR”, “POOR”, “VERY_POOR”.
positiveProtectionsActive	The number of guarantees that were rated positive within the last rating period.

neutralProtectionsActive	The number of guarantees that were rated neutral within the last rating period.
negativeProtectionsActive	The number of guarantees that were rated negative within the last rating period.
RatingPeriodInDays	The period for which the ShopRating is calculated, currently reviews of the last 365 days are considered.
ReviewCountActive	The number of reviews within the last rating period, see RatingPeriodInDays.
ReviewCountTotal	The total number of reviews since the shop started collecting.
reliabilityRatingCountActive	The number of guarantees within the rating period, see RatingPeriodInDays.
reliabilityRatingCountTotal	The number of guarantees a shop collected in its whole membership lifetime.
reliabilityRating	The average of a shops' positive, negative and neutral protections
reliabilityRatingInPercent	The reliabilityRating as percentage
reliabilityRatingCountActive	The number of guarantees within the rating period, see RatingPeriodInDays.
reliabilityRatingCountTotal	The number of guarantees a shop collected in his whole membership lifetime.
ShopRatingCount	The number of reviews of a certain mark, e.g. 500 reviews with mark 5.
mark	The mark which is counted
markDescription	The mark in words, possible values are: "EXCELLENT", "GOOD", "FAIR", "POOR", "VERY_POOR".
countActive	The number of reviews within the rating period
countTotal	The number of reviews within the shops' whole membership lifetime
ShopRatingCritierion	A review consists of several ShopRatingCriterion which represent a CriterionType (Goods, Delivery, Service). Consumers can rate these critieria on a scale from 1-5.
mark	Every ReviewCriterion type can be rated. Consumers can rate each criterion with a mark on a scale from 1-5.
markDescription	The mark in words, possible values are: "EXCELLENT", "GOOD", "FAIR", "POOR", "VERY_POOR".
countActive	The number of reviews within the rating period
countTotal	The number of reviews within the shops' whole membership lifetime

5. Interface Design

5.1 Authentication

Authentication is done via HTTP Basic Authentication. Your login credentials will be sent to you by mail first, after which they will be accessible in the Trusted Shops System backend.

5.2 Accessing Resources

Currently, only the entity Review is available via Trusted Shops' RESTful API. A GET-Request to the following URL will result in a list of reviews order by date in JSON format.

Replace TSID with your Trusted Shops ID. It can be found in your shops' certificate.

GET Request for Reviews

```
GET https://api.trustedshops.com/rest/restricted/v1/shops/TSID/reviews.json
```

5.3 Filters

Filters are appended to the URL as query string parameters and are therefore freely combinable. It is highly recommended to use filters in order to reduce the set of returned elements.

For example, you can narrow down the results for one month:

Example for using the date filters

```
GET …/shops/TSID/reviews.json?startDate=2014-01-01&endDate=2014-01-31
```

When displaying reviews in a frontend application, you might find it helpful to use paging. The following will display a result set of the latest 20 reviews:

Example for the usage of paging filters

```
GET …/shops/TSID/reviews.json?page=0&size=20
```

If you are looking for reviews within a certain mark range, you can use the betterThan and worseThan filters. When using both, they will be combined with a logical AND. The following will output any review with a mark average 3.0 >= x <= 4.0.

Example for the usage of paging filters

```
GET …/shops/TSID/reviews.json?worseThan=4&betterThan=3
```

Filter	Description	Default	Format	Example
startDate	The date to start with	-	yyyy-MM-dd	2014-01-03
endDate	The date until Reviews will be delivered	today	yyyy-MM-dd	2014-02-03

betterThan	Supply a mark from 1-5 to filter Reviews with a mark equal to or greater than the given value.	-	Integer	3
worseThan	Supply a mark from 1-5 to filter Reviews with a mark equal to or lower than the given value.	-	Integer	4
page	For paging support	0	Integer	0
size	Determines how many results will be shown on a single page. Maximum is 100, can be increased indivually on demand.	10	Integer	10

5.4 Versioning

The API version is part of the URL:

API Version as part of the URL

```
GET https://api.trustedshops.com/rest/restricted/v1/shops/TSID/reviews.json
```

The version number is therefore mandatory and cannot be left out. In this way, you can always make sure that you are using the right version of the API.

5.5 Content Negotiation

Available response formats are JSON and XML.

5.5.1 Via file extension

Content negotiation can be done via file extensions in the requested URL:

JSON Response format

```
GET https://api.trustedshops.com/rest/restricted/v1/shops/TSID/reviews.json
```

XML Response format

```
GET https://api.trustedshops.com/rest/restricted/v1/shops/TSID/reviews.xml
```

5.5.2 Via HTTP Accept Header

If you prefer sending HTTP Accept Header for content negotiation, you can just leave the file extension out and send one of the following headers alongside your request.

JSON Response format

```
GET https://api.trustedshops.com/rest/restricted/v1/shops/TSID/reviews
Accept application/json
```

XML Response format

```
GET https://api.trustedshops.com/rest/restricted/v1/shops/TSID/reviews
Accept application/xml
```

Please note that the trailing slash of the URL must be left out, otherwise the API responds with a 404 Not Found.

6. Response Design

6.1 Response Wrapper

The payload of the API will be wrapped, in order to exchange application specific reponse codes, messages and the response status.

Response wrapper in JSON

```
{
   "response":{
      "code":"200",
      "data":{
         …
      }
   },
   "message":"SUCCESS",
   "status":"SUCCESS"
}
```

Response wrapper in XML

```
<response>
    <code>200</code>
    <data>
        …
    </data>
    <message>SUCCESS</message>
    <status>SUCCESS</status>
</response>
```

6.2 Error Codes

When implementing the client, please consider the following list of error codes and their corresponding payloads. Although some might result in a HTTP 200 OK, there is still something wrong with the request. Currently, the response wrapper for these error messages is only available in JSON format.

200 OK

```
{"response":{"code":"400","message":"TS_ID_INVALID","status":"ERROR"}}
```

200 OK

```
{"response":{"code":"403","message":"RESTRICTED_CONTENT_EXCLUDED","status":
"ERROR"}}
```

200 OK

{"response":{"code":"404","message":"TS_ID_NOT_FOUND","status":"ERROR"}}
400 Bad Request
{"response":{"code":"400","message":"BAD_REQUEST","status":"ERROR"}}
403 Forbidden
{"response":{"code":"403","message":"FORBIDDEN","status":"FAIL"}}
404 Not found
{"response":{"code":"404","message":"NOT_FOUND","status":"FAIL"}}
500 Internal Server Error
{"response":{"code":"500","message":"INTERNAL_SERVER_ERROR","status":"FAIL"}}

6.3 Example Responses

Example Response in JSON

```
{
    "response": {
        "code": "200",
        "data": {
            "shop": {
                "tsId": "X12345678910111213141516171819202l2",
                "url": "www.example.com",
                "languageISO2": "en",
                "targetMarketISO3": "GBR",
                "review": [
                    {
                        "changeDate": "2014-01-13T14:46:48+01:00",
                        "comment": "First comment",
                        "confirmationDate": "2014-01-14T02:10:59+01:00",
                        "creationDate": "2014-01-13T14:46:48+01:00",
                        "criterion": [
                            {
                                "mark": "5",
                                "markDescription": "EXCELLENT",
                                "type": "DELIVERY"
                            },
                            {
                                "mark": "5",
                                "markDescription": "EXCELLENT",
                                "type": "GOODS"
                            },
                            {
                                "mark": "5",
                                "markDescription": "EXCELLENT",
                                "type": "SERVICE"
                            }
                        ],
                        "email": "test@trustedshops.de",
                        "mark": "5.00",
                        "markDescription": "EXCELLENT",
                        "orderReference": "45545245",
                        "UID": ""
                    },
                    {
                        "changeDate": "2014-01-13T14:12:43+01:00",
                        "comment": "Second comment",
                        "confirmationDate": "2014-01-14T02:10:59+01:00",
                        "creationDate": "2014-01-13T14:12:43+01:00",
                        "criterion": [
                            {
                                "mark": "5",
                                "markDescription": "EXCELLENT",
                                "type": "DELIVERY"
                            },
                            {
                                "mark": "5",
                                "markDescription": "EXCELLENT",
                                "type": "GOODS"
                            },
                            {
                                "mark": "5",
                                "markDescription": "EXCELLENT",
                                "type": "SERVICE"
                            }
                        ],
                        "email": "test2@trustedshops.com",
                        "mark": "5.00",
                        "markDescription": "EXCELLENT",
                        "orderReference": "36636",
                        "UID": ""
                    },
                    {
                        "changeDate": "2013-12-04T16:01:28+01:00",
                        "comment": "Third comment",
                        "confirmationDate": "2013-12-07T00:12:41+01:00",
                        "creationDate": "2013-12-04T16:01:28+01:00",
                        "criterion": {
                            "mark": "4",
                            "markDescription": "GOOD",
                            "type": "GOODS"
                        },
                        "email": "test2@trustedshops.fr",
                        "mark": "4.00",
                        "markDescription": "GOOD",
                        "orderReference": "123456789",
                        "UID": ""
                    }
                ]
            }
        },
        "message": "SUCCESS",
        "status": "SUCCESS"
    }
}
```

Response wrapper in XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response>
    <code>200</code>
    <data>
        <shop>
```

```
            <tsId>X1234567891011121314151617181920212</tsId>
            <url>www.example.com</url>
            <languageISO2>en</languageISO2>
            <targetMarketISO3>GBR</targetMarketISO3>
            <review>
                <changeDate>2014-01-13T14:46:48+01:00</changeDate>
                <comment>First comment</comment>
                <confirmationDate>2014-01-
14T02:10:59+01:00</confirmationDate>
                <creationDate>2014-01-13T14:46:48+01:00</creationDate>
                <criterion>
                    <mark>5</mark>
                    <markDescription>EXCELLENT</markDescription>
                    <type>DELIVERY</type>
                </criterion>
                <criterion>
                    <mark>5</mark>
                    <markDescription>EXCELLENT</markDescription>
                    <type>GOODS</type>
                </criterion>
                <criterion>
                    <mark>5</mark>
                    <markDescription>EXCELLENT</markDescription>
                    <type>SERVICE</type>
                </criterion>
                <email>test1@trustedshops.de</email>
                <mark>5.00</mark>
                <markDescription>EXCELLENT</markDescription>
                <orderReference>45545245</orderReference>
                <UID></UID>
            </review>
            <review>
                <changeDate>2014-01-13T14:12:43+01:00</changeDate>
                <comment>Second comment</comment>
                <confirmationDate>2014-01-
14T02:10:59+01:00</confirmationDate>
                <creationDate>2014-01-13T14:12:43+01:00</creationDate>
                <criterion>
                    <mark>5</mark>
                    <markDescription>EXCELLENT</markDescription>
                    <type>DELIVERY</type>
                </criterion>
                <criterion>
                    <mark>5</mark>
                    <markDescription>EXCELLENT</markDescription>
                    <type>GOODS</type>
                </criterion>
                <criterion>
                    <mark>5</mark>
                    <markDescription>EXCELLENT</markDescription>
                    <type>SERVICE</type>
                </criterion>
                <email>test2@trustedshops.com</email>
                <mark>5.00</mark>
                <markDescription>EXCELLENT</markDescription>
                <orderReference>36636</orderReference>
                <UID></UID>
            </review>
            <review>
```

```
                <changeDate>2013-12-04T16:01:28+01:00</changeDate>
                <comment>Third comment</comment>
                <confirmationDate>2013-12-
07T00:12:41+01:00</confirmationDate>
                <creationDate>2013-12-04T16:01:28+01:00</creationDate>
                <criterion>
                    <mark>4</mark>
                    <markDescription>GOOD</markDescription>
                    <type>GOODS</type>
                </criterion>
                <email>test3@trustedshops.fr</email>
                <mark>4.00</mark>
                <markDescription>GOOD</markDescription>
                <orderReference>123456789</orderReference>
                <UID></UID>
            </review>
        </shop>
    </data>
    <message>SUCCESS</message>
    <status>SUCCESS</status>
</response>
```