

Our REST API lets you access to resources and perform actions on behalf of a store. For more information, read our [integration guidelines](#) and [FAQs](#), or contact [partnerships@judge.me](mailto:partnerships@judge.me) for help.

## Authentication

Judge.me uses OAuth2 to grant App Developers access to Judge.me API. You need to use the OAuth api\_token generated from the Judge.me app [following this guide](#). Example of how to authenticate with OAuth:

```
GET https://app.judge.me/oauth/authorize?client_id=[your_client_id] & redirect_ur
```

## Judge.me ID vs External ID

You can the resource from our API using two types of id:

- `id`: ID of the resource identified in Judge.me.
- `external_id`: ID of the resource identified in the e-commerce platform (e.g. Shopify). Refer to each endpoint's document to see which type of id is used in each case.

In Judge.me's responses and in query parameters, `id` is Judge.me internal ID and `external_id` is the external ID of the resource.

Meanwhile, in the request body of your update/create requests', `id` is the external ID of the resource.

## Find by ID or other fields

In many endpoints, we allow to find a resource not only by `id`, but also by other fields (e.g. product's handle, reviewer's email, `external_id`). See each endpoint's document for more details.

When `id` is a required param (usually required in the request URL), and you want to find by another field, please set `id` to any negative number,-1, and then set value to the other field by which you want to find.

## Main Widgets

In this section, you can find main [Judge.me widgets](#) used to display Judge.me published reviews on the storefront. Response of the following widget endpoints contains the complete, sanitized, XSS-safe HTML for each widget, that can be embedded inside a storefront page.

## Product Review Widget

Review Widget is usually placed at the bottom of each product page, displaying all reviews of a product. You can use product handle, external ID, or [Judge.me](#) internal ID to specify the product.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

#### QUERY PARAMETERS

---

handle

string

Example: `handle=example-product`

---

external\_id

number

Example: `external_id=999999`

External (Shopify) ID of the Product. Note: this is NOT internal Product ID of Judge.me.

---

id

number (internal\_id)

Example: `id=111`

Judge.me internal ID of the Product

---

page

number

Example: `page=1`

For pagination

---

per\_page

number

Example: `per_page=5`

For pagination

---

## Responses

### — 200

GET /widgets/product\_review

---

## Product Preview Badge

Preview Badge is usually placed below product titles on product pages or inside product thumbnails on collection pages. This widget display the average star rating and review count of each product. You can use product handle, external ID, or [Judge.me](#) internal ID to specify the product.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

#### QUERY PARAMETERS

---

handle

string

Example: `handle=example-product`

---

external\_id

number

Example: `external_id=999999`

External (Shopify) ID of the Product. Note: this is NOT internal Product ID of Judge.me.

---

id

number (internal\_id)

Example: `id=111`

Judge.me internal ID of the Product

---

## Responses

### — 200

GET /widgets/preview\_badge

---

## Featured Reviews Carousel

Reviews Carousel is usually placed on the homepage to showcase specific reviews featured by the store.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

## Responses

### — 200 Successful operation

GET /widgets/featured\_carousel

---

## Floating Reviews Tab

Floating Reviews Tab display all product and store reviews via a floating button on any pages.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

QUERY PARAMETERS

page

number

Example: `page=1`

For pagination

---

per\_page

number

Example: `per_page=5`

For pagination

---

review\_type

string

Enum: `"product-reviews"` `"shop-reviews"`

Example: `review_type=product-reviews`

Whether to show product reviews, or shop reviews, default to product reviews

---

## Responses

### — 200

GET /widgets/reviews\_tab

---

## All Reviews Page

All Reviews Page is a dedicated page to showcase all product and store reviews all in one place.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

### QUERY PARAMETERS

---

page

number

Example: `page=1`

For pagination

---

review\_type

string

Enum: `"product-reviews"` `"shop-reviews"`

Example: `review_type=product-reviews`

Whether to show product reviews, or shop reviews, default to product reviews

---

## Responses

## — 200 Successful operation

GET /widgets/all\_reviews\_page

---

## Verified Reviews Count Badge

Verified Reviews Count Badge displays the number of verified published reviews. Stores need at least 20 verified reviews to use this widget.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

## Responses

### — 200 Successful operation

GET /widgets/verified\_badge

---

## Other Widgets

In this section, you can find other widget information that is not commonly used alone but can be combined in different ways to make a more meaningful widget.

## Total number of all reviews

Return a single total number of product and store reviews.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

## Responses

### — 200 Successful operation

GET /widgets/all\_reviews\_count

---

## Average rating of all reviews

Return a single number of average rating of product reviews and store reviews.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

---

## Responses

### — 200 Successful operation

```
GET /widgets/all_reviews_rating
```

---

## Total number of shop-level reviews

Return a single total number of store reviews.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

---

## Responses

### — 200 Successful operation

```
GET /widgets/shop_reviews_count
```

---

## Average rating of shop-level reviews

Return a single number of average rating of store reviews.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

---

## Responses

### — 200 Successful operation

```
GET /widgets/shop_reviews_rating
```

---

## Widget settings of the shop

Return widget settings of the shop, under HTML format, containing a `<script>` tag and a `<style>` tag. This contains values of widget customization in Judge.me such as text and color, which helps you display the widget correctly.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

### Responses

#### — 200 Successful operation

GET /widgets/settings

## Miracle HTML

Return special HTML that helps show essential parts of widgets before the JS and CSS files are loaded.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

### Responses

#### — 200 Successful operation

GET /widgets/html\_miracle

## Product Checkout Comments widget

Return Checkout Comments widget for a product (for Checkout Comments app only). You can use product handle, external ID, or Judge.me internal ID to specify the product.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or (*PublicAPIKey* and *ShopDomain*) or...

### QUERY PARAMETERS

handle string

Example: `handle=example-product`

external\_id

number

Example: `external_id=999999`

External (Shopify) ID of the Product. Note: this is NOT internal Product ID of Judge.me.

---

id

number (internal\_id)

Example: `id=111`

Judge.me internal ID of the Product

---

## Responses

### — 200

GET /widgets/checkout\_comments\_widget

---

## Reviews

You can use the reviews endpoints to access review information. Common use cases include:

- Synchronize and display reviews in your admin dashboard.
- Get event of new reviews (via webhook) to perform an action on your side.
- Let users manage reviews (publish/hide) on your side. \*Note: these endpoints respond **raw** review information, which may include **unpublished reviews**, or review content that is **not sanitized** yet (so risks of XSS). To render review content on storefront, please use widget endpoints instead.

## Index

Get info of reviews of a product. If `product_id` is not provided, return all product and store reviews of that store.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

### QUERY PARAMETERS

---

per\_page

number

Example: `per_page=10`

For pagination

---

page

number

Example: `page=1`

For pagination

---

reviewer\_id

number (internal\_id)

Example: `reviewer_id=111`

Judge.me internal ID of the Reviewer

---

product\_id

number (internal\_id)

Example: `product_id=111`

Judge.me internal ID of the Product

---

rating

number

Enum: `1` `2` `3` `4` `5`

Example: `rating=5`

Only include reviews with this rating

---

## Responses

### > 200 Array of reviews

GET /reviews

## Response samples

200

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
    "current_page": 1,  
    "per_page": 10,  
    - "reviews": [  
        + { ... }  
    ]  
}
```

## Create

Create a web review in background, similar to submitting a review via the public form on product pages (no authorization required). This endpoint doesn't create any review if the store [disables web reviews](#).

REQUEST BODY SCHEMA: `application/json`

**required**

shop_domain <small>required</small>	string Domain of the shop
platform <small>required</small>	string Enum: <code>"shopify"</code> <code>"woocommerce"</code> <code>"bigcommerce"</code> Platform of the shop
name <small>required</small>	string (person_name)
email <small>required</small>	string <email> (email)
rating <small>required</small>	number (rating) Enum: <code>1</code> <code>2</code> <code>3</code> <code>4</code> <code>5</code>
body <small>required</small>	string
id	number External (Shopify) ID of the Product. Note: this is NOT internal Product ID of Judge.me. If blank, the review is considered as a shop-level review.
reviewer_name_format	string Enum: <code>""</code> <code>"last_initial"</code> <code>"all_initials"</code> <code>"anonymous"</code> How the <code>reviewers</code> want to show their name publicly. Default is showing full name. <code>Stores</code> can also choose how they want to display reviewer's names in all reviews on their stores. When showing reviews, Judge.me will choose the most restrictive option between reviewer's option and store's option. For example, if a reviewer set their display format as all_initials while the store sets the format as last_initial, we will show the reviewer's name as all initials, which is J.S., not John S..
title	string
cf_answers >	Array of objects Answers to the shop's custom forms
picture_urls	Array of strings
ip_addr	string IP address of the reviewer, used to display location of the review. If blank, Judge.me will use IP address that makes the API request

## Responses

## — 200

POST /reviews

### Request samples

#### Payload

##### Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
    "shop_domain": "example.myshopify.com",  
    "platform": "shopify",  
    "id": 999999,  
    "email": "john@example.com",  
    "name": "John Smith",  
    "reviewer_name_format": "",  
    "rating": 5,  
    "title": "Review title",  
    "body": "Review body",  
    - "cf_answers": [  
        + { ... },  
        + { ... },  
        + { ... },  
        + { ... }  
    ],  
    - "picture_urls": [  
        "https://pub-images.judge.me/judgeme/funny/big-pr.jpg",  
        + { ... }  
    ],  
    "ip_addr": "123.123.123.123"  
}
```

---

## Update

Publish or hide a [Judge.me](#) review via your app interface. For authenticity reason, we don't support editing reviews via API.

[AUTHORIZATIONS](#) > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

[PATH PARAMETERS](#)

---

id  
**required**

number  
Example: -1

Judge.me internal ID of the Review

---

REQUEST BODY SCHEMA: application/json

**required**

---

curated

string

value is either ok or spam: ok is for publish reviews, spam is for hide review

---

## Responses

> **200** message info

> **422** message info

PUT reviews/{id}

## Request samples

**Payload**

Content type

application/json

{

  "curated": "ok"

}

Copy

## Response samples

**200**

**422**

Content type

application/json

{

  "message": "Action performed successful"

}

Copy

## Count

Get count of reviews for a specific product or reviewer. If product\_id is not provided, return the count of all product and store reviews of that store.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

### QUERY PARAMETERS

---

reviewer\_id number (internal\_id)

Example: `reviewer_id=111`

Judge.me internal ID of the Reviewer

---

product\_id number (internal\_id)

Example: `product_id=111`

Judge.me internal ID of the Product

---

rating number

Enum: `1` `2` `3` `4` `5`

Example: `rating=5`

Only include reviews with this rating

---

## Responses

### — 200

GET /reviews/count

---

## Get

Get info of a specific review.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

### PATH PARAMETERS

---

id number  
required

Example: `-1`

Judge.me internal ID of the Review

---

## Responses

### > 200 Review info

GET /reviews/{id}

## Response samples

200

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
  - "review": {  
      "id": 111,  
      "title": "Raw review title. This is not sanitized yet so do not print it out  
      "body": "Raw review body. This is not sanitized yet so do not print it out a  
      "rating": 5,  
      "pinned": false,  
      "product_external_id": 999999,  
      "product_title": "Example product",  
      "product_handle": "example-product",  
      + "reviewer": { ... },  
      "source": "email",  
      "curated": "ok",  
      "hidden": false,  
      "verified": "buyer",  
      "created_at": "2019-08-24T14:15:22Z",  
      "updated_at": "2019-08-24T14:15:22Z",  
      "ip_address": "string",  
      "has_published_pictures": false,  
      "has_published_videos": false,  
      + "pictures": [ ... ]  
    }  
}
```



## Webhooks

Subscribe to an event happens in Judge.me. Judge.me will send a POST request to the registered URL containing relevant information for each event.

## Common webhook keys:

1. **review/created** or **review/created\_fail**: to know when a review is created, or not.
2. **review/updated**: to know when a review is updated in Judge.me. In particular, when a review is:
  - curated or mass curated
  - pinned/featured in carousel
  - moved to another product
  - edited from admin or user profile
  - verified review via request emails
  - added/hidden/shown review photos from admin or user profile
3. Widgets update webhooks (e.g. **widget/settings/updated**): to know when Judge.me updates a widget. **\*Note:** You can learn how to verify webhooks from Judge.me following this [guide](#).

## Index

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain* and *OAuthAPIKey*)

## Responses

### > 200 Array of webhooks

GET /webhooks

## Response samples

200

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
    "shop_id": 111,  
    "shop_domain": "example.myshopify.com",  
    - "webhooks": [  
        + { ... }  
    ]  
}
```

## Delete

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain* and *OAuthAPIKey*)

REQUEST BODY SCHEMA: application/json

required

key                    string (webhook\_key)  
 Enum: "widget/settings/updated"  
 "widget/review\_widget/updated"  
 "widget/preview\_badge/updated"  
 "widget/all\_reviews\_count/updated"  
 "widget/all\_reviews\_rating/updated"  
 "widget/shop\_reviews\_count/updated"  
 "widget/shop\_reviews\_rating/updated"  
 "widget/verified\_badge/updated"  
 "widget/all\_reviews\_page/updated"  
 "widget/featured\_carousel/updated"  
 "widget/reviews\_tab/updated" | "widget/html\_miracle/updated"  
 "widget/product\_comment/updated" | "review/created"  
 "review/updated" | "review/created\_fail"

url                    string (webhook\_url)

## Responses

> 200 {

DELETE /webhooks

## Request samples

**Payload**

Content type

application/json

Copy

```
{
  "key": "review/created",
  "url": "https://example.com/webhook/example.myshopify.com/settings/updated"
}
```

## Response samples

**200**

Content type  
application/json

{ }

[Copy](#)

## Create

Create a webhook in Judge.me with a `key` and a `url`. When an event associated with `key` happens, Judge.me will send a POST request to the webhook's `url`.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain* and *OAuthAPIKey*)

REQUEST BODY SCHEMA: application/json  
**required**

webhook > object (RequestWebhook)

After a webhook consumer (i.e. a 3rd-party app) create a webhook in Judge.me with a `key` and a `url`, Judge.me will send a POST request to the webhook's `url` when an event associated with `key` happens.

## Responses

> 200

POST /webhooks

### Request samples

**Payload**

Content type  
application/json

{  
  - "webhook": {  
    "key": "review/created",  
    "url": "<https://example.com/webhook/example.myshopify.com/settings/updated>"  
  }  
}

[Copy](#) [Expand all](#) [Collapse all](#)

## Response samples

**200****Content type**

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
    "shop_id": 111,  
    "shop_domain": "example.myshopify.com",  
    - "webhook": {  
        "id": 111,  
        "failure_count": 0,  
        "key": "review/created",  
        "url": "https://example.com/webhook/example.myshopify.com/settings/updated"  
    }  
}
```

---

## Get

---

**AUTHORIZATIONS:** > (*PrivateAPIKey* and *ShopDomain* and *OAuthAPIKey*)

---

**PATH PARAMETERS****id**  
**required**

number

Example: 

Judge.me internal ID of the Webhook

---

## Responses

**> 200**

GET /webhooks/{id}

## Response samples

**200****Content type**

application/json

```
{
  "shop_id": 111,
  "shop_domain": "example.myshopify.com",
  - "webhook": {
    "id": 111,
    "failure_count": 0,
    "key": "review/created",
    "url": "https://example.com/webhook/example.myshopify.com/settings/updated"
  }
}
```

## Update

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain* and *OAuthAPIKey*)

### PATH PARAMETERS

`id` number  
`required` Example: `-1`  
 Judge.me internal ID of the Webhook

REQUEST BODY SCHEMA: `application/json`

`required`

`webhook` > object (RequestWebhook)  
 After a webhook consumer (i.e. a 3rd-party app) create a webhook in Judge.me with a `key` and a `url`, Judge.me will send a POST request to the webhook's `url` when an event associated with `key` happens.

## Responses

### > 200

PUT /webhooks/{id}

## Request samples

### Payload

Content type  
`application/json`

[Copy](#)
[Expand all](#)
[Collapse all](#)

```
{
  - "webhook": {
    "key": "review/created",
    "url": "https://example.com/webhook/example.myshopify.com/settings/updated"
  }
}
```

## Response samples

**200**

Content type

application/json

[Copy](#)
[Expand all](#)
[Collapse all](#)

```
{
  "shop_id": 111,
  "shop_domain": "example.myshopify.com",
  - "webhook": {
    "id": 111,
    "failure_count": 0,
    "key": "review/created",
    "url": "https://example.com/webhook/example.myshopify.com/settings/updated"
  }
}
```

## Bulk Create

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain* and *OAuthAPIKey*)

REQUEST BODY SCHEMA: application/json

**required**

webhooks > Array of objects (RequestWebhook)

## Responses

— 200

POST /webhooks/bulk\_create

## Request samples

### Payload

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
  - "webhooks": [  
    + { ... }  
  ]  
}
```

## Reviewers

### Get

Get information of the reviewers such as name and email. This is useful when you are building automated flow for users to follow up with reviewers.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

#### PATH PARAMETERS

`id`

**required**

number

Example: `-1`

Judge.me internal ID of the Reviewer. Set to `-1` and use `external_id` or `email` param to find Reviewer by external (Shopify) ID or by email of the Reviewer.

#### QUERY PARAMETERS

`external_id`

number (`external_id`)

Example: `external_id=999999`

External (Shopify) ID of the Reviewer. Note: this is NOT internal Reviewer ID of Judge.me.

`email`

string `<email>` (`email`)

Example: `email=john@example.com`

## Responses

## > 200 Reviewer info

GET /reviewers/{id}

### Response samples

200

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
  - "reviewer": {  
      "id": 111,  
      "email": "john@example.com",  
      "name": "John Smith",  
      "phone": "+123456789",  
      + "tags": [ ... ],  
      "accepts_marketing": true,  
      "unsubscribed_at": "2020-02-20T20:20:20+00:00",  
      "external_id": 999999  
    }  
}
```

## Update

Create or update a reviewer via your app interface.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

### PATH PARAMETERS

id

required

number

Example: `-1`

Judge.me internal ID of the Reviewer. Set to `-1` and use `external_id` or `email` param to find Reviewer by external (Shopify) ID or by email of the Reviewer.

### QUERY PARAMETERS

external\_id

number (external\_id)

Example: `external_id=999999`

External (Shopify) ID of the Reviewer. Note: this is NOT internal Reviewer ID of Judge.me.

email

string &lt;email&gt; (email)

Example: `email=john@example.com`REQUEST BODY SCHEMA: `application/json`required

reviewer &gt;

object (RequestReviewer)

## Responses

### > 200 Reviewer info

PUT /reviewers/{id}

## Request samples

### Payload

Content type

application/json

Copy Expand all Collapse all

```
{
  - "reviewer": {
      "id": 999999,
      "email": "john@example.com",
      "name": "John Smith",
      "phone": "+123456789",
      "tags": "tag1, tag2, tag3",
      "accepts_marketing": true,
      "unsubscribed_at": "2020-02-20T20:20:20+00:00"
    }
}
```

## Response samples

### 200

Content type  
application/json

Copy    Expand all    Collapse all

```
{  
  - "reviewer": {  
      "id": 111,  
      "email": "john@example.com",  
      "name": "John Smith",  
      "phone": "+123456789",  
      + "tags": [ ... ],  
      "accepts_marketing": true,  
      "unsubscribed_at": "2020-02-20T20:20:20+00:00",  
      "external_id": 999999  
    }  
}
```

---

## Data Request

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

REQUEST BODY SCHEMA: application/json  
**required**

object

## Responses

### — 200

POST /reviewers/data\_request

## Request samples

**Payload**

Content type  
application/json

Copy    Expand all    Collapse all

```
{  
  - "customer": {  
      "email": "john@example.com"  
    },  
  - "orders_requested": [  
    "order_external_id_1",  
    "order_external_id_2"  
  ]  
}
```

---

## Shops

### Get current shop's info

Get the basic information of the store such as [Judge.me](#) plan, owner name, email, e-commerce platform, etc. This is helpful when you are developing your app for specific segment of users (e.g. your integration is only available to Judge.me Awesome users).

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

REQUEST BODY SCHEMA: application/json

required

---

object

## Responses

### — 200 Successful operation

GET /shops/info

## Request samples

**Payload**

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{
  - "shop": {
    "id": "111",
    "awesome": "true",
    "country": "US",
    "created_at": "2022-04-18T10:43:52+00:00",
    "currency": "USD",
    "custom_domain": "{{shop_domain}}",
    "dmarc_ok": "true",
    "domain": "{{shop_domain}}",
    "email": "john@example.com",
    "name": "Example store",
    "onboarding_progress": "skip-installation",
    "owner": "John Smith",
    "phone": "44123456789",
    "plan": "basic",
    "platform": "shopify",
    "timezone": "(GMT-05:00) America/New_York",
    "updated_at": "2023-11-23T13:50:40+00:00",
    "widget_version": "3.0"
  }
}
```

## Update current shop

Update store information

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

REQUEST BODY SCHEMA: application/json

**required**

New information of the shop

string

## Responses

### — 200 Successful operation

PUT /shops

## Request samples

## Payload

Content type

application/json

Copy

```
{  
    "domain": "{{shop_domain}}",  
    "custom_domain": "{{shop_domain}}",  
    "email": "john@example.com",  
    "owner": "John Smith",  
    "phone": "+123456789",  
    "name": "Shop Name",  
    "country": "US",  
    "timezone": "(GMT-05:00) America/New_York",  
    "plan": "basic"  
}
```

---

## Uninstall shop from Judge.me

Uninstall the store from Judge.me

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain* and *OAuthAPIKey*)

---

## Responses

— **200** Uninstalled shop successfully

DELETE /shops

---

## Settings

## Index

Get multiple settings values of the store in [Judge.me](#), which can serve as conditions for your app integrations.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

---

## QUERY PARAMETERS

---

`setting_keys[]`

Array of strings

Example:

```
setting_keys[] = admin_email & setting_keys[] = autopublish & setting_keys[] = widget_star_color & setting_keys[] = enable_review_pictures
```

Array of keys of settings

## Responses

&gt; 200

GET /settings

### Response samples

**200**

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{
  - "settings": {
      "admin_email": "support@judge.me",
      "autopublish": true,
      "widget_star_color": "#339999",
      "enable_review_pictures": true
    }
}
```

## Replies

### Create

Create a reply to a Judge.me review on the public Judge.me review widget via your app interface.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

REQUEST BODY SCHEMA: application/json

**required**

review_id <small>required</small>	number (internal_id) Judge.me internal ID of the resource. Note: this ID is from Judge.me internally.
reply > <small>required</small>	object
send_reply_email	boolean To indicate whether to send email to reviewer. Default is <code>true</code>

## Responses

— **200** Reply was created successfully

> **422** Reply was created unsuccessfully

POST /replies

## Request samples

### Payload

Content type

application/json

[Copy](#) [Expand all](#) [Collapse all](#)

```
{  
  "review_id": 111,  
  "send_reply_email": false,  
  - "reply": {  
    "content": "Thanks for the review! We are glad that you like it."  
  }  
}
```

## Response samples

### 422

Content type

application/json

[Copy](#)

```
{  
    "error": "error message"  
}
```

## Private Replies

### Create

Create a private email reply to a Judge.me review privately via your app interface.

AUTHORIZATIONS: > (*PrivateAPIKey* and *ShopDomain*) or *OAuthAPIKey*

REQUEST BODY SCHEMA: application/json

required

review_id <u>required</u>	number (internal_id) Judge.me internal ID of the resource. Note: this ID is from Judge.me internally.
private_reply > <u>required</u>	object
send_private_email	boolean To indicate whether to send email to reviewer. Default is true

### Responses

— **200** Private reply was created successfully

> **422** Private reply was created unsuccessfully

POST /private\_replies

### Request samples

**Payload**

Content type  
application/json

Copy    Expand all    Collapse all

```
{  
    "review_id": 111,  
    "send_private_email": false,  
    - "private_reply": {  
        "email_subject": "email subject",  
        "email_body": "email body"  
    }  
}
```

## Response samples

**422**

Content type  
application/json

Copy

```
{  
    "error": "error message"  
}
```