Direct Freight API

Direct Freight API description

Introduction

In order to use the Direct Freight API please first contact us to get an API Token. This teken needs to be sent as a HTTP header argument in every request to our API.

api-token="SECRET KEY"

Please keep your api-token a secret. If the token is compromised please contact us to get a new one.

This api token authenticates your application and gives it access to the API.

The API token should not be confused with the end-user-token. That token is given to you via the end_user_authentications endpoint when the user logs in. This token is needed in order to use certain features, and get certain sensitive information.

Both Tokens are stateless, and delivered via the HTTP headers.

HTTP Response Code

HTTP Response code will vary based on the operation completed, or error that occurred. Below are the codes you will see from Direct Freight.

Code	Meaning	Parameters
200	OK	Response to a successful request. Usually a GET, PUT, or PATCH.
201	Created	Response to a POST. Will also have a link to the new information if applicable.
204	No Content	This is the response for a successful DELETE. There will be no body.
400	Bad Request	This will be returned if the request was incorrectly formed. Syntax error, or missing parameters. Wil return an error message.
401	Unauthorized	It will be returned when requests are made to things with none or insufficient permissions.
404	Not Found	When a non-existent resource is requested.
422	Unprocessable Entity	If the API processes the request but finds validation errors. For example a invalid city, or negative radius. It will return an error message.
429	Too Many Requests	When a request is rejected due to rate limiting.

End User Authentications

GET /end_user_authentications/ [#api-end_user_authentications-get_auth_status]	Get current authentication level for the user.
DELETE /end_user_authentications/ [#api-end_user_authentications-logout]	Logs a user out.
POST /end_user_authentications/ [#api-end_user_authentications-login]	Logs a user in.
GET /end_user_authentications/passwords/ [#api-end_user_authentications-email_password]	Request a forgotten password.
PATCH /end_user_authentications/passwords/ [#api-end_user_authentications-update_password]	Update login password.

GET /end_user_authentications/dates/ [#api-end_user_authentications-get_date]	Gets an authoritative date_time string.
POST /end_user_authentication/users/ [#api-end_user_authentications-create_user]	Create a new user.

GET /end_user_authentications/

Get current authentication level for the user.

Header

Field	Type	Description
`end-user-token`	string	The logged in user's token.

Success 200

Field	Type	Description
`status`	string	Will return the logged in user's status. Valid status's are "subscribed", "trial", or "overdue".

- Return their Status: [#success-examples-end_user_authentications-get_auth_status-0_0_0-0]

```
staus: "subscribed"
```

DELETE /end_user_authentications/

Logs a user out.

This logs a user out. the end-user-token will no longer be valid.

Header

Field	Type	Description
`end-user-token`	string	The logged in user's token.

POST /end_user_authentications/

Logs a user in.

This function sends login credentials, and retrieves a stateless token.

Parameter

Field	Type	Description

Field	Type	Description
realm	string	This specificies who to log in with. If left blank we will assume email/password. Valid realms are username, email, facebook(not implemneted), google(not implemented), and linkedin(not implemented).
login	string	The e-mail, or username information needed to log in.
secret	string	The password, or token required for this login.

Success 200

Field	Type	Description
end-user-token	string	A generated tonken to authenticatate as this user.

- Example token: [#success-examples-end_user_authentications-login-0_0_0-0]

```
end-user-token: "XX423ASOEUHT4AESUTH"
```

GET /end_user_authentications/passwords/

Request a forgotten password.

Parameter

Field	Type	Description
email	string	The user e-mail to send a password to if valid.

PATCH /end_user_authentications/passwords/

Update login password.

Header

Field	Type	Description
end-user-token	string	The logged in user's token.

Parameter

Field	Type	Description
old_password	string	Their old password.

Field	Type	Description
`new_password`	string	Their new password.

GET /end_user_authentications/dates/

Gets an authoritative date_time string.

Success 200

Field	Type	Description
`now`	string	The current date and time based on our server.

POST /end_user_authentication/users/

Create a new user.

Parameter

Field	Type	Description
`email`	string	This must be a valid e-mail.
`password`	string	Currently this is a 6-10 digit alphanumeric password.

Success 200

Field	Type	Description
`end-user-token`	string	The account is created, and a generated tonken to authenticatate as this user is returned.

- Example token: [#success-examples-end_user_authentications-create_user-0_0_0-0]

```
end-user-token: "XX423ASOEUHT4AESUTH"
```

Error Codes

Fill this in with common error codes as they appear. Examples might be.

Code	Error	Description
001	Generic Validation error.	The error body will explain which parameter failed validation and how.
002	Login Failure.	Bad email or password.