HelpDeskZ

Release 2.0.1

Andres Mendoza

Mar 09, 2021

WELCOME TO HELPDESKZ

1 Welcome to HelpDeskZ **1**

2 Server Requirements **3**

3 Fresh Installation **5**
- 3.1 Configuration file . . . 5
- 3.2 Install it! . . . 5

4 Upgrading from previous version **7**
- 4.1 Upgrading from 1.x to 2.0 . . . 7
- 4.2 Upgrading from 2.0 to 2.0.1 . . . 8
- 4.3 Upgrading from 2.0.1 to 2.0.2 . . . 8

5 HelpDeskZ Repositories **9**

6 Email Piping **11**
- 6.1 Email configuration . . . 11
- 6.2 Email forwarding . . . 11

7 IMAP Fetching **13**
- 7.1 Email configuration . . . 13
- 7.2 Cron configuration . . . 13

8 API activation **15**
- 8.1 Authentication and authorization . . . 15

9 Users **17**
- 9.1 Create a new user . . . 17
- 9.2 Retrieve a list of all users . . . 18
- 9.3 Retrieve details of user by ID . . . 19
- 9.4 Update user account . . . 19
- 9.5 Delete user account . . . 20

10 Departments **21**
- 10.1 Create a new department . . . 21
- 10.2 Retrieve a list of all departments . . . 22
- 10.3 Retrieve details of department by ID . . . 23
- 10.4 Update department . . . 23
- 10.5 Delete department . . . 24

11 Support Tickets **27**

11.1 Create a new ticket . . . 27
11.2 Retrieve a list of all tickets . . . 28
11.3 Retrieve details of ticket by ID . . . 29
11.4 Update ticket . . . 30
11.5 Delete department . . . 31

12 Ticket Messages **33**
12.1 Add a new message . . . 33
12.2 Retrieve messages from a ticket . . . 34

13 Attachments **37**
13.1 Retrieve list of attachments . . . 37
13.2 Retrieve file content from an attachment . . . 38
13.3 Delete attachment . . . 39

14 Staff **41**
14.1 Retrieve a list of all staff users . . . 41
14.2 Retrieve details of staff user by ID . . . 42
14.3 Staff Authentication . . . 42

HTTP Routing Table **45**

CHAPTER

ONE

WELCOME TO HELPDESKZ

HelpDeskZ is a free PHP based software which allows you to manage your site's support with a web-based support ticket system.

HelpDeskZ is developed with CodeIgniter PHP framework 4

CHAPTER

TWO

SERVER REQUIREMENTS

- PHP version 7.3 or newer is required
- **intl** extension, **mbstring** extension and **imap** extension installed.
- MySQL (5.1+) via the *MySQLi* driver

CHAPTER

THREE

FRESH INSTALLATION

Note: Be sure that your server meets the *HelpDeskZ requirements*.

3.1 Configuration file

- Unzip the HelpDeskZ script package.
- Browse to directory **/hdz/app/Config/** and rename the file **Helpdesk.new.php** to **Helpdesk.php**.
- Edit this file and complete the required information (Site URL, database information, etc).

3.2 Install it!

- Connect with FTP to the *public folder* of your server.
- Upload all HelpDeskZ in the directory that you will install it (domain directory or subdomain).
- Open **/install** in your browser, for example http://support.mysite.com/install (modify your URL).
- The HelpDeskZ setup script will run. Click INSTALL HELPDESKZ and follow the instructions through installation wizard.
- Now it's time to setup your help desk! Open the staff panel in your browser, for example: http://support.mysite.com/staff Use the login details that you entered in the installation process.
- Take some time and get familiar with all the available settings. Most should be self-explanatory.
- Thanks for using HelpDeskZ!

Note: If you want to make a new installation again, then be sure you removed the file **/hdz/writable/cache/instal.config** to unlock the installation wizard.

CHAPTER

FOUR

UPGRADING FROM PREVIOUS VERSION

Please read the upgrade notes corresponding to the version you are upgrading from.

4.1 Upgrading from 1.x to 2.0

Note: v2.x was rewritten from scratch so the attachments from tickets will be lost in this new version.

The upgrade process is very simple and just follow these steps:

4.1.1 Configuration file

- Unzip the HelpDeskZ script package.
- Browse to directory **/hdz/app/Config/** and rename the file **Helpdesk.new.php** to **Helpdesk.php**.
- Edit this file and complete the required information (Site URL, database information, etc).

4.1.2 Update your site

- Upload all files of v2.x to your HelpDeskZ directory.
- Open **/install** in your browser, for example http://support.mysite.com/install (modify your URL).
- The HelpDeskZ setup script will run. Click UPGRADE HELPDESKZ and follow the instructions through upgrade wizard.
- Now it's time to setup your help desk! Open the staff panel in your browser, for example: http://support.mysite.com/staff
- Take some time and get familiar with all the available settings. Most should be self-explanatory.
- Thanks for using HelpDeskZ!

4.2 Upgrading from 2.0 to 2.0.1

The upgrade process is very simple and just upload these files:

- /index.php
- /hdz/*.*

4.2.1 Update your site

- Open **/install** in your browser, for example http://support.mysite.com/install (modify your URL).
- The HelpDeskZ setup script will run. Click `Upgrade my HelpDeskZ´ and follow the instructions through upgrade wizard.
- Thanks for using HelpDeskZ!

4.3 Upgrading from 2.0.1 to 2.0.2

4.3.1 Changelog

- Framework update to work with PHP >= 7.3.
- The auto-close of tickets is fixed.
- The staff notification when new ticket is opened is fixed.
- Ticket notes added.

4.3.2 Upgrade process

If you made custom modifications in your template for client panel, then make a backup of the folder **/hdz/app/Views/client/** The upgrade process is very simple and just upload these files:

- /index.php
- /.htaccess *[required if you have issues visiting pages in your site]*
- /hdz/app/*.*
- /hdz/framework/*.*
- /hdz/install/*.*

4.3.3 Update your site

- Open **/install** in your browser, for example http://support.mysite.com/install (modify your URL).
- The HelpDeskZ setup script will run. Click `Upgrade my HelpDeskZ´ and follow the instructions through upgrade wizard.
- Restore the template files of client panel.
- Thanks for using HelpDeskZ!

CHAPTER

FIVE

HELPDESKZ REPOSITORIES

The HelpDeskZ is an **open source** project and repository is hosted in Github

CHAPTER

SIX

EMAIL PIPING

HelpDeskZ supports email piping, this allows the auto-creation of tickets from incoming emails to a set email address.

6.1 Email configuration

- In Staff Panel, go to Setup -> Email addresses.
- Add a new email address or edit the email for Piping configuraiton.
- Go to **Incoming** tab and select **Pipe**

6.2 Email forwarding

- In your hosting panel, go to email forwarding.
- Enter the email address that you configured in your staff panel.
- For destination, select Pipe
- Enter the path to pipe.php, for example /public_html/helpdeskz/pipe.php

Note: To make it work correctly, verify that pipe.php has executable permissions (CHMOD 755)

CHAPTER

SEVEN

IMAP FETCHING

HelpDeskZ supports IMAP fetching, this allows the auto-creation of tickets from incoming emails to a set email address.

7.1 Email configuration

- In Staff Panel, go to Setup -> Email addresses.
- Add a new email address or edit the email for IMAP configuration.
- Go to **Incoming** tab and select **IMAP**
- Complete your configuration with the Host, Port, Username and Password

7.2 Cron configuration

- Add a cron job to execute your command every minute or 5 minutes. It depends the timeframe you want to check emails.
- Command: **/usr/local/bin/php /PATH_TO_HELPDESK/index.php imap_fetcher**

Note: Many hosts uses the IMAP port 993 by default

CHAPTER

EIGHT

API ACTIVATION

- In Staff Panel, go to Setup -> API configuraiton.
- Create a new API.
- Select the permissions for your new API.
- Enter the IP that will connect with your API.
- After creation, the system will generate your API token.

8.1 Authentication and authorization

All endpoints require authentication.

8.1.1 Token

The `Authorization` HTTP header can be specified with `Token: <your-access-token>`.

CHAPTER

NINE

USERS

9.1 Create a new user

POST /api/users/create

Query Parameters

- **fullname** (*string*) – Client's Full Name
- **email** (*string*) – Client's Email address
- **notify** (*boolean*) – 1 = Client will receive an email with login information

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/users/ \
-F 'fullname="John Doe"' \
-F 'email="john.doe@demo.com"' \
-F 'notify="0"'
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'http://helpdeskz.web/api/users/create',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => array('fullname' => 'John Doe','email' => 'john.doe@demo.
↪com','notify' => '1'),
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "user_id": 1,
```

(continues on next page)

(continued from previous page)

```
    "message": "User account was created."
}
```

9.2 Retrieve a list of all users

GET /api/users/

Query Parameters

- **email** (*string*) – Find a client by email
- **page** (*numeric*) – Page query is used to view next page

Example request:

cURL

```
curl \
-H 'Token: <token>' https://demo.helpdeskz.com/api/users
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/users',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "total_users": 2,
    "total_pages": 1,
    "users": [
        {
            "id": "2",
            "fullname": "John Doe",
            "email": "john.doe@demo.com"
        },
        {
            "id": "1",
            "fullname": "John Doe",
            "email": "john.doe123@demo.com"
        }
    ]
}
```

9.3 Retrieve details of user by ID

GET /api/users/show/<user_id>

Example request:

cURL

```
curl \
-H 'Token: <token>' https://demo.helpdeskz.com/api/users/show/1
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/users/show/1',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "user_data": {
        "id": "1",
        "fullname": "John Doe",
        "email": "john.doe@demo.com"
    }
}
```

9.4 Update user account

POST /api/users/update/<user_id>

Query Parameters

- **new_email** (*string*) – New client's email address

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/users/update/1 \
-F 'new_email="john.doe123@demo.com"'
```

PHP

```php
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/users/update/1',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => array('new_email' => 'john.doe123@demo.com'),
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "message": "Email was changed."
}
```

9.5 Delete user account

POST /api/users/delete/<user_id>

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/users/delete/1
```

PHP

```php
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/users/delete/1',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "message": "Account was removed."
}
```

CHAPTER

TEN

DEPARTMENTS

10.1 Create a new department

POST /api/departments/create

Query Parameters

- **name** (*string*) – Department name
- **private** (*boolean*) – 0=public department, 1=private department

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/departments/create/ \
-F 'name="Bug report"' \
-F 'private="0"'
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/departments/create/',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => array('name' => 'Bug report','private' => '0'),
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "department_id": 4,
    "message": "Department was created."
}
```

10.2 Retrieve a list of all departments

GET /api/departments/

Example request:

cURL

```
curl \
-H 'Token: <token>' https://demo.helpdeskz.com/api/departments
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/departments',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "departments": [
        {
            "id": "1",
            "name": "General",
            "private": "0"
        },
        {
            "id": "2",
            "name": "Advertising",
            "private": "0"
        },
        {
            "id": "3",
            "name": "Sales",
            "private": "0"
        },
        {
            "id": "4",
            "name": "Bug report",
            "private": "0"
        }
    ]
}
```

10.3 Retrieve details of department by ID

GET /api/departments/show/<user_id>

Example request:

cURL

```
curl \
-H 'Token: <token>' https://demo.helpdeskz.com/api/departments/show/4
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/departments/show/4',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "department": {
        "id": "4",
        "name": "Bug report",
        "private": "0"
    }
}
```

10.4 Update department

POST /api/departments/update/<department_id>

Query Parameters

- **name** (*string*) – New department name
- **private** (*boolean*) – 0=public department, 1=private department

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/departments/update/4 \
-F 'name="Issues report"'
-F 'private="0"'
```

PHP

```php
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/departments/update/1',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => array('name' => 'Issues report', 'private' => '0'),
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "message": "Department was updated."
}
```

10.5 Delete department

POST /api/departments/delete/<department_id>

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/departments/delete/4
```

PHP

```php
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/departments/delete/4',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "message": "Department and its tickets were removed."
}
```

Note: With this action, all tickets from this department will be removed.

CHAPTER

ELEVEN

SUPPORT TICKETS

11.1 Create a new ticket

POST /api/tickets/create

Query Parameters

- **opener** (*string*) – [user,staff] The person who is opening this ticket
- **user_id** (*numeric*) – User ID
- **staff_id** (*numeric*) – Staff ID, it is required if opener is `staff´
- **department_id** (*numeric*) – Department ID
- **subject** (*string*) – Ticket subject
- **body** (*string*) – Ticket message
- **attachment[]** (*file*) – (optional) Attachment file
- **notify** (*boolean*) – 1=User will receive an email with ticket information

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/tickets/create/ \
-F 'opener="staff"' \
-F 'user_id="1"' \
-F 'staff_id="1"' \
-F 'department_id="4"' \
-F 'subject="Hello world"' \
-F 'body="This is a test message."' \
-F 'attachment[]=@"/home/andres/Images/hdz.png"' \
-F 'notify="0"'
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'ttps://demo.helpdeskz.com/api/tickets/create/',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => array('opener' => 'user','user_id' => '1','staff_id' => '1
↪','department_id' => '4','subject' => 'Hello world','body' => 'This is a test
↪message.','attachment[]'=> new CURLFILE('/home/andres/Images/hdz.png'),'notify'␣
↪=> '0'),
```

(continues on next page)

(continued from previous page)

```
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "ticket_id": 1,
    "message": "Ticket was created."
}
```

11.2 Retrieve a list of all tickets

GET /api/tickets/

Query Parameters

- **department_id** (*numeric*) – (optional) Department ID
- **user_id** (*numeric*) – (optional) User ID
- **status_id** (*numeric*) – (optional) 1: Open, 2: Answered, 3: Awaiting reply, 4: In progress, 5: Closed

Example request:

cURL

```
curl \
-H 'Token: <token>' https://demo.helpdeskz.com/api/tickets
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/tickets',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "total_tickets": 1,
    "pages": 1,
```

(continues on next page)

(continued from previous page)


```
    "tickets": [
        {
            "id": "1",
            "user_id": "1",
            "department_id": "4",
            "subject": "Hello world",
            "date": "1611619574",
            "last_update": "1611619936",
            "status": "4",
            "replies": "0",
            "user_fullname": "John Doe",
            "department_name": "Issues report"
        }
    ]
}
```


11.3 Retrieve details of ticket by ID

GET /api/tickets/show/<ticket_id>

Example request:

cURL

```
curl \
-H 'Token: <token>' https://demo.helpdeskz.com/api/tickets/show/1
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/tickets/show/1',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:


```
{
    "success": 1,
    "ticket": {
        "id": "1",
        "user_id": "1",
        "department_id": "4",
        "subject": "Hello world",
        "date": "1611619574",
        "last_update": "1611619574",
        "status": "1",
        "replies": "0",
```


(continues on next page)

(continued from previous page)

```
        "user_fullname": "John Doe",
        "department_name": "Issues report"
    }
}
```

11.4 Update ticket

POST /api/tickets/update/<ticket_id>

Query Parameters

- **department_id** (*numeric*) – (optional) Department ID
- **status_id** (*numeric*) – (optional) 1: Open, 2: Answered, 3: Awaiting reply, 4: In progress, 5: Closed

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/tickets/update/1 \
-F 'name="Issues report"'
-F 'private="0"'
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/departments/update/1',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => array('name' => 'Issues report', 'private' => '0'),
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "message": "Ticket was updated."
}
```

11.5 Delete department

POST /api/tickets/delete/<ticket_id>

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/tickets/delete/1
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/tickets/delete/1',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "message": "Ticket was removed."
}
```

Note: With this action, ticket and its messages and attachments will be be removed.

CHAPTER

TWELVE

TICKET MESSAGES

12.1 Add a new message

POST /api/messages/create

Query Parameters

- **ticket_id** (*numeric*) – Ticket ID
- **replier** (*string*) – [staff,user] The person who is replying this ticket
- **staff_id** (*numeric*) – Staff ID, required only if replier is `staff´
- **message** (*string*) – Message content
- **attachment[]** (*file*) – (optional) Attachment file
- **close** (*boolean*) – 1=Close ticket after reply, this is only valid if replier is `staff´

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/messages/create/ \
-F 'ticket_id="1"' \
-F 'replier="staff"' \
-F 'staff_id="1"' \
-F 'message="Answering a ticket."' \
-F 'attachment[]=@"/home/andres/Images/homebg.jpg"' \
-F 'close="0"'
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'ttps://demo.helpdeskz.com/api/messages/create/',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => array('ticket_id' => '1','replier' => 'staff','staff_id'␣
↪=> '1','message' => 'Answering a ticket','attachment[]'=> new CURLFILE('/home/
↪andres/Images/homebg.jpg'),'close' => '0'),
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
```

(continues on next page)

(continued from previous page)

```
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "message": "Message was created and added to ticket."
}
```

12.2 Retrieve messages from a ticket

GET /api/messages/show/<ticket_id>/

Example request:

cURL

```
curl \
-H 'Token: <token>' https://demo.helpdeskz.com/api/messages/show/1
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/messages/show/1',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "pages": 1,
    "total_replies": 3,
    "messages": [
        {
            "id": "3",
            "date": "1611619936",
            "customer": "1",
            "staff_id": "0",
            "message": "This is user answer"
        },
        {
            "id": "2",
            "date": "1611619889",
            "customer": "0",
```

(continues on next page)

(continued from previous page)


```
            "staff_id": "1",
            "message": "Answering a ticket"
        },
        {
            "id": "1",
            "date": "1611619574",
            "customer": "1",
            "staff_id": "0",
            "message": "This is a test message."
        }
    ]
}
```

CHAPTER

THIRTEEN

ATTACHMENTS

13.1 Retrieve list of attachments

GET /api/attachments/

Query Parameters

- **ticket_id** (*numeric*) – (optional) Ticket ID
- **msg_id** (*numeric*) – (optional) Message ID

Example request:

cURL

```
curl \
-H 'Token: <token>' https://demo.helpdeskz.com/api/attachments/
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/attachments/',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "files": [
        {
            "id": "2",
            "name": "homebg.jpg",
            "filetype": "image/jpeg",
            "msg_id": "2",
            "size": "116748"
        },
```

(continues on next page)

(continued from previous page)

```
        {
            "id": "3",
            "name": "homebg.jpg",
            "filetype": "image/jpeg",
            "msg_id": "3",
            "size": "116748"
        }
    ]
}
```

13.2 Retrieve file content from an attachment

GET /api/attachments/show/<attachment_id>

Example request:

cURL

```
curl \
-H 'Token: <token>' https://demo.helpdeskz.com/api/attachments/show/1
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/attachments/show/1',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "name": "hdz.png",
    "filetype": "image/png",
    "size": "13497",
    "content":
↪"iVBORw0KGgoAAAANSUhEUgAAAmIAAACKCAYAAAAJ3iYgAAAACXBIWXMAAA7DAAAOwwHHb6hkAAAAGXRFWHRTb2Z0d2FyZ
↪DPuTPZCJvgAiKyuKNoUXCtmgRUCKBfF/
↪BnW61LW2qr1GIpIEnmTDbQtrZi1eJStWqtoZtCQGQVcQe1LiCIiggGFcIaSDIz9/z+mNiisuS5c2/
↪uLOf9evF6icy5zyFkJuc+93nOQ1BKKaVUUuNFBe2Du2iwBatIIP0BHAugI4AOPqeWuQQ7QbQDkNWAvEtkLbayaN7EC+ZvM
↪c5HHVAjCP+WGH5fMnLha60J0EJMKaWUSjKVcy/sSdHY7wG5DPqzOkVRrU3yi9LihR/
↪u91VtlY7KLLyoIDenIXj4pBHzP/
↪I7F6VUeqqsrOwVjUZPJKL+ItKemUv9zskNVbMLr4FN94DQ3u9cVMIaReTWkhGL7t3XC4JtmY1KT7c/
↪fcHhkazYaWSjnxBOJKAfduEkm+xlAL7rd35KqdQ2derUTs3NzUfbtn0igNMA9ANwSjQaPQQARAQA3gaQ0oUYM1tZpy/
↪5AwQ36zRJ2sglonsqZxeeHs2TH3Hh4ug3X6CFmGo1XlSQG2gIfteiWH+IdRIIJwPSL4pYOxIApFOsSil3MHM3AA8DOKmxs
↪PxGjNbwUFLHoHgar9zUe4joR9m7aaDeFHB5d8sxrQQU62W0xjob5M9L15xid/pKKXSW2cAQ/
↪1Ooq1kD1zyO4EWYWlNcHHWLusBANft+b8tn9JRSimlFIDKWUWjhHCL33moNnFtde3gMXv+Dy3ElFJKKZ/
↪cUVvQjQj3+52HajsC+X3FMxf2+er3WoqppZRSPokqcDvij2FV5sizArHff/UbLcSUUkopH1TMLjoKkO/
↪7nYfyg1xSWVswENBCTCmllPJFwMYtAAJ+56H8YQndAmghppRSSrU5rhmVLRau8jsP5R8huV12p87
↪poO0rlMoQzNw5KysrJxKJ5FuWlQcgFwCIaFcsFmsCsDMvL2/3hAkTdvibqVLpL5C35WwIuvqdh/
↪JVu0gwu0gLMaXSRHV1ddfm5uYTABwP4BgAPQAcCeDwlv/
↪OjUQiAADbtvd6jd27d4OZdwD4FMA6AOuJaIWILM/Ly3tTizSl3GFZdoHfOSj/
↪EagwWD27qMYkyAY9XVK84AmvkjqQqtrCSUQ0oLWvFxvrJo9Y+Csvc1KqLU2fPj2rrq7uRACnAhgA4BQAJzQ3Nx/
↪s0hAdED9Cph/
↪w3+NjsHv3bpuZVwFYQkQzO3TosHDcuHG7XRpTqUzT3+8ElP8I6B8UwSijIBJfD3EWoe8CKG7t64nwjofpKNWmmPnlurq6A
```

(continues on next page)

(continued from previous page)

```
}
```

13.3 Delete attachment

POST /api/attachments/delete/<attachment_id>

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/attachments/delete/1
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/attachments/delete/1',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "message": "File was removed from servers."
}
```

CHAPTER

FOURTEEN

STAFF

14.1 Retrieve a list of all staff users

GET /api/staff/

Query Parameters

- **username** (*string*) – Find a user by username

Example request:

cURL

```
curl \
-H 'Token: <token>' https://demo.helpdeskz.com/api/staff
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/staff',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "users": [
        {
            "id": "1",
            "username": "admin",
            "fullname": "Andres Mendoza",
            "email": "andres@demo.com",
            "registration": "1611613586",
            "last_login": "1611618058"
        }
    ]
}
```

14.2 Retrieve details of staff user by ID

GET /api/staff/show/<staff_id>

Example request:

cURL

```
curl \
-H 'Token: <token>' https://demo.helpdeskz.com/api/staff/show/1
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/staff/show/1',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'GET',
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "staff_data": {
        "id": "1",
        "username": "admin",
        "fullname": "Andres Mendoza",
        "email": "andres@demo.com",
        "registration": "1611613586",
        "last_login": "1611618058"
    }
}
```

14.3 Staff Authentication

POST /api/staff/auth

Query Parameters

- **username** (*string*) – Staff username
- **password** (*string*) – Staff password
- **two_factor** (*numeric*) – Two-Factor Authentication code, this is required if two-factor authentication is active in account
- **ip_address** (*string*) – IP Address of client

Example request:

cURL

```
curl \
-X POST \
-H 'Token: <token>' https://demo.helpdeskz.com/api/staff/auth/ \
-F 'username="admin"' \
-F 'password="demo123"' \
-F 'two_factor="815435"' \
-F 'ip_address="127.0.0.1"'
```

PHP

```
<?php
$curl = curl_init();
curl_setopt_array($curl, array(
  CURLOPT_URL => 'https://demo.helpdeskz.com/api/staff/auth/',
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_CUSTOMREQUEST => 'POST',
  CURLOPT_POSTFIELDS => array('username' => 'admin','password' => 'demo123','two_
↪factor' => '815435','ip_address' => '127.0.0.1'),
  CURLOPT_HTTPHEADER => array(
    'Token: <token>'
  ),
));
$response = curl_exec($curl);
curl_close($curl);
```

Example response:

```
{
    "success": 1,
    "message": "You have been logged in."
}
```

Last revision: January 26th, 2021

HTTP ROUTING TABLE

/api

GET /api/attachments/, 37
GET /api/attachments/show/<attachment_id>, 38
GET /api/departments/, 22
GET /api/departments/show/<user_id>, 23
GET /api/messages/show/<ticket_id>/, 34
GET /api/staff/, 41
GET /api/staff/show/<staff_id>, 42
GET /api/tickets/, 28
GET /api/tickets/show/<ticket_id>, 29
GET /api/users/, 18
GET /api/users/show/<user_id>, 19
POST /api/attachments/delete/<attachment_id>, 39
POST /api/departments/create, 21
POST /api/departments/delete/<department_id>, 24
POST /api/departments/update/<department_id>, 23
POST /api/messages/create, 33
POST /api/staff/auth, 42
POST /api/tickets/create, 27
POST /api/tickets/delete/<ticket_id>, 31
POST /api/tickets/update/<ticket_id>, 30
POST /api/users/create, 17
POST /api/users/delete/<user_id>, 20
POST /api/users/update/<user_id>, 19