Workice CRM Documentation

Release latest

William Mandai

Dec 15, 2019

Installation

This documentation will help you learn all about creating projects, sending invoices, receiving payments, creating tasks, converting estimates to invoices, recurring invoices, creating creditnotes, support system and much more.

Note: Workice CRM is build with Laravel Framework

To purchase Workice CRM Click Here

CHAPTER 1

Install

Thanks for purchasing Workice CRM.

Note: Workice CRM is build with Laravel 6 Framework

Note: Workice requires PHP >= 7.2 and MySQL.

Attention: You can check your server requirements by downloading this file extract it and upload to your server and access it via a URL i.e http://yourdomain.com/requirements.php (Delete after use).

1.1 Required PHP Extensions

- OpenSSL
- PDO
- Mbstring
- Tokenizer
- JSON
- CURL
- Mysqli
- IMAP
- Zip
- GD

Release Notes: Changelogs

Any feature you need to see on Workice? request it here

1.2 Create Database

> **Attention:** Do not use a password that contains a #(Hash) character (It will be treated as a comment)

You'll need to create a new database along with a user to access it. Most hosting companies provide an interface to handle this or you can run the SQL statements below.

- Give your database a name e.g **workicecrm**
- Create a database user and set up a password
- Add the user to the database and give the user **All privileges** to the database

```
CREATE DATABASE workicecrm;
CREATE USER 'workicecrm' IDENTIFIED BY 'workicecrm@@';
GRANT ALL PRIVILEGES ON workicecrm.* to workicecrm@'%' identified by 'workicecrm@@';
FLUSH PRIVILEGES;
```

- You will need to enter this credentials in the steps below.

There are 2 steps to install Workice CRM

- Using the built in web installer
- Artisan install method (Requires terminal)

> **Attention:** Do not modify config files located in /config folder incase you need to modify configuration paramers do it in .env file (Located in ROOT/Main folder). When you modify /config files your changes may be overwritten during an update.

1.3 Web Installer

Download the code from Envato Market Downloads page. The zipped file includes all dependencies. The following steps shows how to install Workice CRM;

The steps below shows how you can install workice to be accessible via a sub domain e.g **https://crm.yourdomain.com**

- Create a folder inside **public_html or www** e.g **workice** and upload your downloaded files into the folder **/public_html/workice** (Make sure it includes hidden files e.g .env)
- Go to **sub domains** on your cpanel dashboard and create a sub domain
- Enter your preferred sub domain e.g crm.yourdomain.com
- Enter **/public_html/workice** as your sub domain **Document Root**
- Save it and access **http://crm.yourdomain.com/installer**

> **Attention:** The url **/installer** does not point to a folder so do not attempt to create a folder named installer

If you encounter an issue displaying images refer to troubleshooting tips below.

1.3.1 Installation

Once you can access the site the initial setup screen will enable you to configure the database and email settings as well as create the initial admin user. The first page of the web installer checks if your server meets the requirements for Workice to run.

> **Attention:** Workice requires PHP 7.2+

Click **Next** if everything is alright if an extension is missing please contact your hosting provider or install it. The next step checks **directory permissions**. The folders listed should be writable please do NOT set your permissions to **777**. The next step requires database and account information. Enter the information and complete the installation.

> **Attention:** You will need to setup CRON to run every minute as shown below otherwise invoices/estimates will not be sent or calculated.

1.3.2 File Permissions

The webserver should be able to write to this directories **storage**, **public** and **bootstrap/cache**. Here is a sample of how you can set the permissions in ubuntu server.

```
sudo chown -R ubuntu:www-data /path/to/workice
cd /path/to/workice
sudo find -type f -exec chmod 664 {} \;
sudo find -type d -exec chmod 775 {} \;
sudo chgrp -R www-data bootstrap/cache storage
sudo chmod -R ug+rwx bootstrap/cache storage
```

- Enter your application name and application URL (e.g https://crm.yourdomain.com)
- Enter your database access information that you used when creating database.
- Enter your admin account information. (This is the admin account you are going to login with)
- Click on install and Workice will perform the migrations and seeding.
- If everything went well, you should get a success screen. Click on **Exit** and login using admin account you created above.

> **Attention:** You will need to setup email inorder to verify users accounts. More on that in next article (Configure)

1.4 Installing through SSH (Artisan command)

If you need to install Workice CRM using `php artisan` command proceed as follows;

- Open **.env** file and update your database credentials i.e **DB host,DB User etc** (You can change other configurations later).
- Run command `php artisan workice:install` to start the installation.
- You will be asked to enter admin email and password.
- After successfull install you can now access your dashboard using http://crm.yourdomain.com
- Use your admin account to login.

Note: Admin account created using `php artisan workice:install` command does not require email verification.

1.5 Email Configuration

- Workice CRM supports SMTP, Mailgun, Postmark, SparkPost, Amazon SES, and sendmail.
- If you have no idea how to configure email sending, read on the next guide **Configuration**.
- For more information check https://discuss.workice.com/

1.6 CRON Configuration

Add a CRON job as shown below;

```
* * * * * cd /path/to/workice && php artisan schedule:run >> /dev/null 2>&1
```

This Cron will call Workice command scheduler every minute. When the **schedule:run** command is executed, Workice will evaluate your scheduled tasks and runs the tasks that are due.

More information available here https://discuss.workice.com

1.7 Queue Configuration (optional)

Note: For VPS or AWS EC2 users, we recommend installing Supervisord to monitor your processes. Steps on how to install Supervisor on ubuntu are described below

---

If you need to use supervisord to monitor your queued jobs follow the steps below;

- Open **app/Console/Kernel.php** and comment the line `$schedule->command('queue:work --workicedaemon --queue=default,high,normal,low --tries=3')....`
- Now install/start supervisor as described below;

1.7.1 Installing Supervisor

Supervisor is a process monitor for the Linux operating system, and will automatically restart your queue:work process if it fails. To install Supervisor on Ubuntu, you may use the following command:

```
sudo apt-get install supervisor
```

Supervisor configuration files are typically stored in the **/etc/supervisor/conf.d** directory. Within this directory, you may create any number of configuration files that instruct supervisor how your processes should be monitored. For example, let's create a workice-worker.conf file that starts and monitors a queue:work process:

```
[program:workice-worker]
process_name=%(program_name)s_%(process_num)02d
command=php /path/to/workice/artisan queue:work --queue=default,high,normal,low --
↪tries=3
autostart=true
autorestart=true
user=ubuntu
numprocs=1
redirect_stderr=true
stdout_logfile=/path/to/workice/worker.log
```

You can refer to laravel docs

1.7.2 Starting Supervisor

Once the configuration file has been created, you may update the Supervisor configuration and start the processes using the following commands:

`sudo supervisorctl reread`

`sudo supervisorctl update`

`sudo supervisorctl restart all`

For more information on Supervisor, consult the Supervisor documentation.

See the details here for additional configuration options.

1.8 Troubleshooting

- Check your webserver log (ie, /var/log/apache2/error.log) and the application logs (storage/logs/laravel-error.log) for more details or set `APP_DEBUG=true` in .env
- Getting 404 not found when i access http://crm.mydomain.com/installer - Ensure your sub domain ROOT Document points to /path/to/workice/public folder and not /path/to/workice folder.
- I cannot see a folder named **installer** - The url /installer is a laravel route and not a folder. You will be redirected to /installer if the application detects that the app needs to be installed.
- To resolve `file_put_contents(...): failed to open stream: Permission denied` run `chmod -R 777 storage` then `chmod -R 755 storage`
- Running `composer install --no-dev` and `composer dump-autoload` can sometimes help with composer problems.
- Getting error message "Database connection/migration failed" all database credentials are correct. Check that your database user has enough privileges to perform database actions, workice database should be empty or your password contains a #(Hash).
- Composer install error: `Fatal error: Allowed memory size of...` Try the following: `php -d memory_limit=128M /usr/local/bin/composer install --no-dev`
- If you are installing on PHP 7.3 and encounter an error `preg_match_all(): JIT compilation failed: no more memory` you will need to modify your php.ini file and enter **pcre.jit=0** to fix the issue.

- My images are not loading - May be caused by wrong symlink. Edit public/index.php file and write this line at the top of the code after `<?php` opening tag `symlink('../your-crm-folder/storage/app/public', './storage');`
- My CRONs are not running and i get an error **ErrorException with message 'Invalid argument supplied for foreach()' in /home/project/vendor/symfony/console/Input/ArgvInput.php** to fix this, enter your CRON to run every minute as shown `php -d register_argc_argv=On /path/to/workice/artisan schedule:run >/dev/null`
- Not receiving emails and invoices/estimates balances are not shown correctly. You need to setup CRONS as shown above to fix that.

CHAPTER 2

Configure

You can check .env.example file to see additional settings.

2.1 CRON Setup

Create one CRON that runs **every minute** as follows;

```
* * * * * php /path-to-your-project/artisan schedule:run >/dev/null
```

2.2 Email Queues

Queues allow you to defer the processing of a time consuming task, such as sending an email, until a later time. Deferring these time consuming tasks drastically speeds up web requests to your application. You can change the queue driver in your .env file `QUEUE_DRIVER=database`

Note: You can process the jobs by running `php artisan queue:work --queue=default,high,normal,low --tries=3 --daemon`.

2.2.1 Supervisord Configuration

We highly recommend configuring supervisord to monitor your processes and restart your queue if it fails. Read more on Supervisord Configuration

2.3 Application Configuration

You can set your application to production by modifying this value in .env file.

- `APP_ENV` - Set it to **production**
- `APP_DEBUG` - Set it to **false**

2.4 System Configuration

Here some of the system configurations you can change in your .env

- `EMAIL_TRACKING_ENABLE` - Enables email tracking for leads.
- `PDF_FONT` - Font to use on PDF generated.
- `DAILY_DIGEST_ENABLED` - Enables daily email summary - Default true.
- `DAILY_DIGEST_AT` - Sets the time when daily summary should be sent - Default 23:58
- `ACTIVITY_DAYS` - The number of days activity logs take to be cleared - Default 7 days
- `TICKETS_DUE_DAYS` - Number of days before a ticket expires - Default 3 days
- `TASKS_DUE_DAYS` - Days before a task is overdue - Default 7 days
- `ALERT_TODO_DAYS` - Number of days to notify me before a todo is overdue
- `ESTIMATE_REMIND_DAYS` - Number of days before a reminder is sent to client about expiring estimate. Default 2 days
- `CONTRACT_REMIND_DAYS` - Days before a client is notified of expiring contract that has not been signed. Default 2 days
- `ENABLE_DRIFT` - Enable Drift (Read below on how to configure)
- `ENABLE_CRISP` - Enable Crisp (Read below on how to configure)
- `ENABLE_ONESIGNAL` - Enable Onesignal (Read below on how to configure)
- `ENABLE_TAWK` - Enable Tawk (Read below on how to configure)

2.4.1 Backup Settings

- `BACKUP_DISKS` - Comma separated list of backup disks e.g **local,s3**
- `BACKUPS_MAIL_ALERT` - Email to send notifications on successfull backup
- `BACKUPS_SLACK_WEBHOOK` - Slack webhook to post backup notifications
- `BACKUPS_SLACK_CHANNEL` - Slack channel to post backup notifications. Default blank

2.4.2 Email Settings

- `MAIL_DRIVER` - Email driver. Default **smtp**
- `MAIL_HOST` - Email host e.g **smtp.mailtrap.io**
- `MAIL_PORT` - Outgoing email port e.g **2525** or **465**
- `MAIL_USERNAME` - Outgoing email username
- `MAIL_PASSWORD` - Outgoing email password
- `MAIL_ENCRYPTION` - Default null other options **tls** and **ssl**
- `MAIL_FROM_ADDRESS` - The email address that sends emails (Your company address)

- MAIL_FROM_NAME - Your company name that appears on emails

> **Attention:** To send a test email go to **Settings** -> **System Info** and click on **Test Email** button.

2.4.3 Paypal Configuration

To setup paypal IPN;

- Login to Paypal account
- Set the IPN to https://{YOUR-DOMAIN}/webhook/paypal/ipn replace {YOUR-DOMAIN} with your actual domain e.g https://app.workice.com/webhook/paypal/ipn

> **Attention:** To enable PayPal Live, go to **Settings** -> **Payment Settings**

2.4.4 Stripe Configuration

To configure Stripe, proceed as follows;

- Login to your stripe dashboard account
- Get your API Keys by clicking Developers section
- Obtain your stripe webhook keys by clicking on Webhooks under Developers section of your stripe dashboard.

Open your .env file in Workice CRM and modify the values below;

```
STRIPE_KEY={YOUR_STRIPE_PUBLISHABLE_KEY}
STRIPE_SECRET={YOUR_STRIPE_SECRET_KEY}
STRIPE_WEBHOOK_SECRET={YOUR_STRIPE_WEBHOOK_KEY}
```

2.4.5 Stripe Webhook Configuration

To handle Stripe webhooks, proceed as follows;

- Login to your stripe dashboard and click on Developers section.
- Click Webhooks -> Add Endpoint button
- Enter webhook URL as https://{YOUR-DOMAIN}/stripe/webhook replace {YOUR-DOMAIN} with your actual domain e.g https://app.workice.com/stripe/webhook

By default, Workice CRM will automatically handle cancelling subscriptions that have too many failed charges (as defined by your Stripe settings), customer updates, customer deletions, subscription updates, and credit card changes;

Required stripe webhooks;

- customer.subscription.updated
- customer.subscription.deleted
- customer.updated
- customer.source.deleted
- customer.deleted

2.4.6 Razorpay Configuration

To configure RazorPay, proceed as follows;

- Login to your razorpay dashboard account
- Get your API Keys by clicking Settings -> API Keys section

Open your .env file in Workice CRM and modify the values below;

```
RAZORPAY_KEY={RAZORPAY_KEYID}
RAZORPAY_SECRET={RAZORPAY_SECRET}
```

Attention: Create Razorpay webhook and enter webhook URL as https://{YOUR-DOMAIN}/webhook/razorpay/ipn replace {YOUR-DOMAIN} with your actual domain e.g https://app.workice.com/webhook/razorpay/ipn

2.4.7 Braintree Configuration

To configure Braintree, proceed as follows;

- Login to your braintree dashboard account
- Get your API Keys by clicking Settings -> API section
- Just below the API keys you'll see your Merchant ID

Open your .env file in Workice CRM and modify the values below;

```
BRAINTREE_MERCHANT_ID={BRAINTREE_MERCHANT_ID}
BRAINTREE_PUBLIC_KEY={BRAINTREE_PUBLIC_KEY}
BRAINTREE_PRIVATE_KEY={BRAINTREE_PRIVATE_KEY}
```

Attention: You will need to enter your Merchant Account in Settings -> Payment Settings -> Braintree Merchant Account

Attention: To enable Braintree Live, go to **Settings** -> **Payment Settings**

2.4.8 WePay Configuration

To configure WePay gateway, proceed as follows;

- Login to your WePay dashboard account
- Get your API Keys by clicking on your business account
- Copy and replace the values below with your WePay API Keys

Open your .env file in Workice CRM and modify the values below;

```
WEPAY_ACCOUNT_ID={WEPAY_ACCOUNT_ID}
WEPAY_CLIENT_ID={WEPAY_CLIENT_ID}
WEPAY_SECRET_ID={WEPAY_CLIENT_SECRET}
WEPAY_ACCESS_TOKEN={WEPAY_ACCESS_TOKEN}
```

Attention: To enable WePay Live, go to **Settings -> Payment Settings**

2.4.9 2Checkout Configuration

To configure 2checkout, proceed as follows;

- Login to your 2checkout dashboard account
- Get your API Keys by clicking on API section
- Obtain your SELLER ID by clicking on your 2chekout avatar and copy **Account Number**.

Open your .env file in Workice CRM and modify the values below;

```
2CHECKOUT_PUBLISHABLE_KEY={2CHECKOUT_PUBLISHABLE_KEY}
2CHECKOUT_PRIVATE_KEY={2CHEKOUT_PRIVATE_KEY}
2CHECKOUT_SELLER_ID={2CHEKOUT_SELLER_ID}
```

Attention: To enable 2Checkout Live, go to **Settings -> Payment Settings**

2.4.10 Mollie Configuration

To configure mollie, proceed as follows;

- Login to your Mollie dashboard account
- Get your API Keys by clicking on Developers section

Open your .env file in Workice CRM and modify the values below;

```
MOLLIE_KEY={MOLLIE_API_KEY}
```

2.5 Google Calendar Setup

To display events from your Google Calendar, Go to **Settings -> System Settings** and enter your Google Calendar API key and your Google Calendar ID. Once the settings are configured, your events will display on Workice calendar.

2.6 Crisp, Drift, Onesignal and Tawk.to

After you have signed up to the services above, proceed as follows;

2.6.1 Onesignal

After you have enabled onesignal in your .env, copy the **script code** they have given you and paste it in **/resources/views/partial/onesignal.blade.php** (Replace the sample code in that file).

2.6.2 Drift

After you have enabled drift in your .env, copy the **script code** they have given you and paste it in **/resources/views/partial/drift.blade.php** (Replace the sample code in that file).

2.6.3 Crisp

After you have enabled crisp in your .env, copy the **script code** they have given you and paste it in **/resources/views/partial/crisp.blade.php** (Replace the sample code in that file).

2.6.4 Tawk

After you have enabled tawk.to in your .env, copy the **script code** they have given you and paste it in **/resources/views/partial/tawk.blade.php** (Replace the sample code in that file).

2.7 Google ReCaptcha

To enable recaptcha, first get your recaptcha key and secret from Google. Open **.env** file on the ROOT folder and enter your values as shown.

```
NOCAPTCHA_SECRET=YOUR-RECAPTCHA-SECRET
NOCAPTCHA_SITEKEY=YOUR-RECAPTCHA-SITE-KEY
```

Go to **Settings** > **System Settings** then enable **Use Recaptcha**.

2.8 Using a (Reverse) Proxy

If you need to set a list of trusted (reverse) proxies you can modify **app/Http/Middleware/TrustProxies.php** file. Your trusted proxies should be listed as an array on the **$proxies** property of this middleware. In addition to configuring the trusted proxies, you may configure the proxy **$headers** that should be trusted:

```
protected $proxies = [
      '192.168.1.1',
      '192.168.1.2',
      '10.0.0.0/8',
      '192.168.0.0/16'
 ];
```

CHAPTER 3

Social Logins

In addition to typical, form based authentication, Workice CRM also provides a simple, convenient way to authenticate with OAuth providers.

Tip: Workice CRM currently supports authentication with **Facebook, Twitter, LinkedIn, Google, GitHub and GitLab**

Attention: Before using Social Logins, you will have to enable it in **Settings** > **System Settings**. Activate the **Social Login** checkbox.

Note: New user accounts will be created if they do not exist.

3.1 Facebook Configuration

- Start by creating a Facebook Application on the Developers Console.
- Copy your newly created application **App ID** and **App Secret**
- Open your .env file located in your Root folder and set `FACEBOOK_CLIENT_ID={YOUR-APP-ID}` and `FACEBOOK_CLIENT_SECRET={YOUR-APP-SECRET}`.
- Still in your facebook developer console, under **Valid OAuth Redirect URIs** enter your redirect url as `https://your-domain.com/callback/facebook`. Example; If you have installed workice in https://example.com then the callback url will be `https://example.com/callback/facebook`.
- Now your users can login using Facebook.

3.2 Twitter Configuration

- Start by creating a Twitter Application on the Developers Console.
- Copy your newly created application Consumer API Keys **API key** and **API secret key**
- Open your .env file located in your Root folder and set `TWITTER_CLIENT_ID={YOUR-API-KEY}` and `TWITTER_CLIENT_SECRET={YOUR-API-SECRET}`.
- Still in your twitter developer console, under **Callback URLs** enter your redirect url as `https://your-domain.com/callback/twitter`. Example; If you have installed workice in https://example.com then the callback url will be `https://example.com/callback/twitter`.

3.3 Google Configuration

- Start by creating Google Application on the Developers Console.
- Copy your newly created application **Client ID** and **Client Secret**
- Open your .env file located in your Root folder and set `GOOGLE_CLIENT_ID={YOUR-CLIENT-ID}` and `GOOGLE_CLIENT_SECRET={YOUR-SECRET-KEY}`.
- Still in your google developer console, under **Authorized redirect URIs** enter your redirect url as `https://your-domain.com/callback/google`. Example; If you have installed workice in https://example.com then the callback url will be `https://example.com/callback/google`.

3.4 Github Configuration

- Start by creating Github Application on the Developers Console.
- Copy your newly created application **Client ID** and **Client Secret**
- Open your .env file located in your Root folder and set `GITHUB_CLIENT_ID={YOUR-CLIENT-ID}` and `GITHUB_CLIENT_SECRET={YOUR-SECRET-KEY}`.
- Still in your github developer console, under **Authorization callback URL** enter your redirect url as `https://your-domain.com/callback/github`. Example; If you have installed workice in https://example.com then the callback url will be `https://example.com/callback/github`.

3.5 LinkedIn Configuration

- Start by creating LinkedIn Application on the Developers Console.
- Copy your newly created application **Client ID** and **Client Secret**
- Open your .env file located in your Root folder and set `LINKEDIN_CLIENT_ID={YOUR-CLIENT-ID}` and `LINKEDIN_CLIENT_SECRET={YOUR-SECRET-KEY}`.
- Still in your linkedin developer console, under **Redirect URLs** enter your redirect url as `https://your-domain.com/callback/linkedin`. Example; If you have installed workice in https://example.com then the callback url will be `https://example.com/callback/linkedin`.

3.6 Gitlab Configuration

- Start by creating Gitlab Application in Applications Settings.
- Enter your **Redirect URI**. Example; If you have installed workice in https://example.com then the callback url will be `https://example.com/callback/gitlab`. (Use 1 line per URI)
- Under **Scopes** check the **API** checkbox to enable it
- Copy your newly created application **Application ID** and **Secret**
- Open your .env file located in your Root folder and set `GITLAB_CLIENT_ID={YOUR-APPLICATION-ID}` and `GITLAB_CLIENT_SECRET={YOUR-SECRET-KEY}`.

CHAPTER 4

Update

Note: We recommend backing up your files and database before updating the app.

Workice CRM checks for updates daily and notifies you via email if there is a new update. You can disable update notifications in **Settings** > **System Settings** > **Update Notifications**.

Attention: Updates require a valid purchase code. Enter your purchase code in **Settings** > **System Settings** and save.

To update your CRM got to **Settings > System info** and click on **Schedule Update**. Enter the time when the app should update and timezone.

Attention: Your application will be set to maintenance mode when performing the update and restored after the update process has completed.

If you're moving servers make sure to copy over the .env file.

If incase the system update fails and shows **Maintenance Mode**, run the command `php artisan up` manually and check the storage/logs folder for issues. You will receive an email if the update is successful/fails.

4.1 Fixing Permissions

Never set your folder to 777 permission. If you give any of your folders 777 permissions, you are allowing any user to read, write and execute any file in that directory. What this means is you have given any user (any hacker or malicious person in the entire world) permission to upload ANY file, virus or any other file, and THEN execute that file.

To fix permissions issue run command `sudo chown -R my-user:www-data /path/to/your/workice` The first **my-user** is name of the user, and the second **www-data** is the name of the group.

Then give both yourself and the webserver permissions as shown;

`sudo find /path/to/your/workice -type f -exec chmod 664 {} \; sudo find /path/to/your/workice -type d -exec chmod 775 {} \;`

Then give the webserver the rights to read and write to storage and cache.

`sudo chgrp -R www-data storage bootstrap/cache`

`sudo chmod -R ug+rwx storage bootstrap/cache`

Credits https://stackoverflow.com/questions/30639174/how-to-set-up-file-permissions-for-laravel-5-and-others

CHAPTER 5

Changelogs

5.1 Version 2.1.5 - 15 December 2019

- Added staff timecards
- Added gocardless direct debit
- Added Invoicr PDF engine
- Upload WYSIWYG images to folder instead of base64 encode to database
- Fixed copying project issue
- Fixed Project Template not copying milestones,tasks etc
- Added option to disable discount and tax columns on estimates/invoices
- Fixed bulk invoice payments
- Fixed tasks today count on top header
- Fixed importing csv
- Fixed issue with saving todo
- Show error when invoice overpaid
- Create tickets for only registered users
- Fixed language, permission and translations 404 error
- Added correct cron path on Settings > System Info page
- Added invoice recurring biennially
- Fixed invoice double taxation

5.2 Version 2.1.2 - 25 September 2019

- Added Aircall integration (sync)
- Upgraded to Laravel 6 LTS
- Added html editor for texts (Summernote)
- Added Project Custom Fields
- Added SMS integration (Inbound/Outbound - Twilio)
- Added Bulk SMS to leads
- Updated Stripe Subscriptions (SCA Rules)
- Added option to manually very user accounts
- Changed WhatsApp driver to Wablas
- Added Github Issues integrations with Project Issues
- Added Contract Templates
- Fixed known issues

5.3 Version 2.1.0 - 15 June 2019

- Added WhatsApp integration via ApiWha
- Added WhatsApp Opt-In
- Added unlimited recurring option (without end dates)
- Fixed next recurring dates for recurring invoices,expenses and tasks
- Fixed Onesignal issue
- Added support and lead custom email senders
- Added option to set default application locale
- Added project feedback report
- Send notifications via WhatsApp
- Changed datatables to POST from GET
- Added option to increase number of digits for invoices
- Fixed known issues

5.4 Version 2.0.9 - 28 May 2019

- Added project templates
- Added Pagseguro payment for brazil
- Added more dashboard stats
- Fixed todo list URL
- Added web to lead form

- Fixed font in PDFs
- Added convert deal to project
- Added custom item units
- Fixed attaching projects to tickets
- Added notification to team when task created
- Fixed known issues

5.5 Version 2.0.8 - 13 May 2019

- Fixed invoices and estimates public link responsive
- Fixed Nexmo SMS
- Fixed PDF fonts
- Fixed files not showing in user profile
- Added search contacts
- Fixed adding items conflict in creditnotes
- Fixed calendar locale
- Fixed time format in client dashboard
- Added default company logo if missing
- Fixed stage_id required error when importing deals
- Fixed lead stage calculations
- Fixed invoice auto reminders

5.6 Version 2.0.7 - 08 May 2019

- Fixed quick access issue when task/project deleted.
- Fixed notification not sent to task creator
- Fixed PDF for shared estimates
- Fixed PDF download for contracts by client
- Added paytm for India
- Fixed customer verification
- Made activities clickable
- Show social logins individually
- Fixed send email when ticket status changed
- Fixed files without title not downloadable
- Transfer notes to deals when lead is converted to deal
- Transfer notes to client comments when lead is converted to customer

5.7 Version 2.0.6

- Add button to compute analytics in Reports
- Fixed date issue when importing data from old system
- Fixed email files not setting correct adapter
- Fixed missing translation strings
- Added run cron via URL
- Fixed show task name for running timer

5.8 Version 2.0.5

- Fixed deleting contact issue
- Fixed reset password hashing twice
- Fixed PDF not showing image

5.9 Version 2.0.4

- Fixed archiving projects when importing from freelancer
- Fixed issue in cancelling invoices
- Fixed invoicing project issue
- Fixed error displaying deals in staff dashboard

5.10 Version 2.0.3

- Added artisan commands execution
- Fixed ticket assignment error
- Fixed deleting user observer

5.11 Version 2.0.2

- Fix importing old data
- Fixed installation company name containing white space
- Fixed department filter in tickets
- Fixed Recaptcha

5.12 Version 2.0.1

- Fix help me link not working in dashboard
- Lock exchange rates
- Option to update invoices, estimates, credits and expenses exchange rates
- Fixed knowledgebase rating issue
- Added option to turn off project feedback email
- Fixed Paypal Live IPN issue
- Added invoice overpayment alert

5.13 Version 2.0.0 - 01 May 2019

- Initial release

CHAPTER 6

API

Workice CRM provides a RESTful API. Below are the API Endpoints you can use;

6.1 Authorization

To access the API, you first need to copy your Access Token. Go to top right corner next to your profile avatar image and select **Settings** then click **API Settings**. Click on **Show Access Token** and copy your access token.

Note: Insert your Access Token directly into the authorization header on any API requests. Example below;

```
curl -X GET
-H "Authorization: Bearer <bearer>"
-H "Content-Type: application/json"
"https://{endpoint}/api/v1/invoices"
```

6.2 Authentication Errors

If the bearer token is invalid or expired you will receive a response with the status code set to **HTTP 401 Unauthorized**.

If the access_token is valid but you don't have enough scope to perform this request you will receive a response with the status code set to **HTTP 403 Forbidden**.

6.3 HTTP-Codes

Actions and errors yield different HTTP response codes. Please have a look at the expected response codes in the following list.

- `200` Request OK
- `201` New resource created
- `304` The resource has not been changed
- `400` The request parameters are invalid
- `401` The bearer token or the provided api key is invalid
- `403` You do not possess the required rights to access this resource.
- `404` The resource could not be found / is unknown.
- `415` The data could not be processed or the accept header is invalid.
- `422` Could not save the entity
- `500` An unexpected condition was encountered
- `503` The server is not available (maintenance work).

6.4 HTTP-Headers

Attention: You must provide the following headers in each request:

Header	Valid Values	Description
Accept	application/json	API Format
Authorization	Bearer $AccessToken	Replace with your Access Token

6.5 Invoices

6.5.1 `POST` - /api/v1/invoices

Create new invoice

```
POST /api/v1/invoices
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
reference_no	optional	Invoice reference code
title	optional	Invoice title
client_id	required	Invoice client ID
due_date	required	Invoice due date
currency	required	Invoice currency
notes	optional	Invoice notes
tax	optional	Invoice tax 1 percentage
tax2	optional	Invoice tax 2 percentage
extra_fee	optional	Invoice extra fee percentage
discount	optional	Invoice discount percentage
project_id	optional	Project ID related to invoic
is_visible	optional	Set to 0 to hide invoice from client
line_items[]	optional	Array of invoice items
tags[]	optional	Array list of tags e.g `tags[design]`

6.5.2 `GET` - /api/v1/invoices/{id}

Get invoice information

```
GET /api/v1/invoices/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response


```
{
  "type": "invoices",
  "id": "80",
  "attributes": {
      "id": 80,
      "reference_no": "INV-20181224-0080",
      "title": "Richie Rodriguez Website Project",
      "due_date": "2019-01-05T05:30:21+03:00",
      "tax": "6.21",
      "tax2": "0.00",
      "discount": "9.16",
      "currency": "USD",
      "extra_fee": "0.00",
      "status": "Not Paid",
      "payable": "7508.13",
      "tax_total": "480.43",
      "paid_amount": "0.00",
      "late_fee": "0.00",
      "balance": "7508.13",
      "business": {
          "id": 10,
          "name": "Herzog LLC",
          "contact_person": "muller.william@example.com"
      },
```


(continues on next page)

(continued from previous page)

```
      "created_at": "2018-12-24T05:30:23+03:00",
      "updated_at": "2019-01-03T18:32:05+03:00"
  }
}
```

6.5.3 PUT - /api/v1/invoices/{id}

Update an invoice

```
PUT /api/v1/invoices/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
id	required	Invoice ID
reference_no	optional	Invoice reference code
title	optional	Invoice title
client_id	required	Invoice client ID
due_date	required	Invoice due date
notes	optional	Invoice notes
tax	optional	Invoice tax 1 percentage
tax2	optional	Invoice tax 2 percentage
extra_fee	optional	Invoice extra fee percentage
discount	optional	Invoice discount percentage
project_id	optional	Project ID related to Invoice
is_visible	optional	Set to 0 to hide invoice from client
line_items[]	optional	Array of invoice items

6.5.4 DELETE - /api/v1/invoices/{id}

Delete invoice

```
DELETE /api/v1/invoices/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.5.5 GET - /api/v1/invoices

Get a list of all invoices

```
GET /api/v1/invoices
Accept: application/json
Authorization: Bearer $AccessToken
```

6.5.6 GET - /api/v1/invoices/{id}/payments

Show invoice payments

```
GET /api/v1/invoices/{id}/payments
Accept: application/json
Authorization: Bearer $AccessToken
```

6.5.7 GET - /api/v1/invoices/{id}/comments

Show invoice comments

```
GET /api/v1/invoices/{id}/comments
Accept: application/json
Authorization: Bearer $AccessToken
```

6.5.8 GET - /api/v1/invoices/{id}/items

Show invoice product lines

```
GET /api/v1/invoices/{id}/items
Accept: application/json
Authorization: Bearer $AccessToken
```

6.6 Leads

6.6.1 POST - /api/v1/leads

Create a new lead

```
POST /api/v1/leads
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
email	required	Email address
name	optional	Full Names
source	required	Lead source ID
lead_value	optional	Lead value
stage_id	optional	Lead stage id
sales_rep	optional	User responsible for this lead
job_title	optional	Lead Job Title
company	optional	The company associated with this lead
phone	optional	Lead phone number
address1	optional	Lead address
city	optional	City
state	optional	State
zip_code	optional	Zip Code
country	optional	Country
website	optional	Website URL
skype	optional	Lead skype address
message	optional	Additional lead message
tags[]	optional	Array list of tags e.g `tags[design]`

6.6.2 GET - /api/v1/leads/{id}

Get lead information

```
GET /api/v1/leads/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response

```
{
  "type": "leads",
  "id": "50",
  "attributes": {
      "id": 50,
      "name": "Janet Ruecker PhD",
      "source": {
          "id": 31,
          "name": "Youtube"
      },
      "email": "ymohr@example.net",
      "stage": {
          "id": 24,
          "name": "Contacted"
      },
      "job_title": "Electrical Parts Reconditioner",
      "company": "Bauch, Beahan and Macejkovic",
      "phone": "5056437040843",
      "mobile": null,
```

(continues on next page)

(continued from previous page)

```
        "address": {
            "address1": "2124 Windler Plains Suite 342\nLake Elishire, OK 92549",
            "address2": null,
            "city": "New Daphnee",
            "state": null,
            "zipcode": null,
            "country": "Holy See (Vatican City State)"
        },
        "timezone": null,
        "website": "green.com",
        "social": {
            "skype": null,
            "facebook": null,
            "twitter": null,
            "linkedin": null
        },
        "agent": {
            "id": 3,
            "name": "Lorna Rogahn",
            "email": "luna66@example.net"
        },
        "lead_score": 10,
        "due_date": "2019-01-07T05:30:31+03:00",
        "lead_value": "$30.22",
        "message": null,
        "has_activity": 0,
        "has_email": 0,
        "next_followup": "2018-12-27T05:30:31+03:00",
        "unsubscribed_at": null,
        "archived_at": null,
        "created_at": "2018-12-24T05:30:31+03:00",
        "updated_at": "2018-12-24T05:30:31+03:00"
    }
}
```

6.6.3 PUT - /api/v1/leads/{id}

Update a lead

```
PUT /api/v1/leads/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
email	required	Email address
name	optional	Full Names
source	required	Lead source ID
lead_value	optional	Lead value
stage_id	optional	Lead stage id
sales_rep	required	User responsible for this lead
job_title	optional	Lead Job Title
company	optional	The company associated with this lead
phone	optional	Lead phone number
address1	optional	Lead address
city	optional	City
state	optional	State
zip_code	optional	Zip Code
country	optional	Country
website	optional	Website URL
skype	optional	Lead skype address
message	optional	Additional lead message

6.6.4 `DELETE` - /api/v1/leads/{id}

Delete a lead

```
DELETE /api/v1/leads/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.6.5 `GET` - /api/v1/leads

Get a list of all leads

```
GET /api/v1/leads
Accept: application/json
Authorization: Bearer $AccessToken
```

6.6.6 `GET` - /api/v1/leads/{id}/comments

Show leads comments

```
GET /api/v1/leads/{id}/comments
Accept: application/json
Authorization: Bearer $AccessToken
```

6.6.7 `GET` - /api/v1/leads/{id}/todos

Show lead todo list

```
GET /api/v1/leads/{id}/todos
Accept: application/json
Authorization: Bearer $AccessToken
```

6.6.8 `GET` - /api/v1/leads/{id}/calls

Show lead calls

```
GET /api/v1/leads/{id}/calls
Accept: application/json
Authorization: Bearer $AccessToken
```

6.7 Deals

6.7.1 `POST` - /api/v1/deals

Create a new deal

```
POST /api/v1/deals
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
title	required	Deal title
pipeline	required	Pipeline ID
stage_id	required	Deal stage ID
contact_person	required	Deal contact person
deal_value	optional	Deal value e.g 500
due_date	optional	The date a deal is expected to close
organization	optional	Deal Organization
status	optional	Deal status. Default open
source	optional	Deal source
currency	optional	Deal Currency
tags[]	optional	Array list of tags e.g `tags[design]`

6.7.2 `GET` - /api/v1/deals/{id}

Get deal information

```
GET /api/v1/deals/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response


```
{
  "type": "deals",
  "id": "1",
  "attributes": {
      "id": 1,
      "title": "Boyle-Konopelski Deal",
      "stage": {
          "id": 5,
          "name": "Needs Analysis"
      },
      "currency": "USD",
      "deal_value": "875.55",
      "contact_person": {
          "id": 1,
          "email": "muller.william@example.com"
      },
      "organization": {
          "id": 4,
          "name": "Kertzmann, Schoen and Zboncak",
          "email": "wkub@example.org"
      },
      "due_date": null,
      "status": "open",
      "won_time": null,
      "lost_time": null,
      "lost_reason": null,
      "source": {
          "id": 28,
          "name": "Web"
      },
      "pipeline": {
          "id": 20,
          "name": "Delivery"
      },
      "user_id": 1,
      "next_followup": null,
      "archived_at": null,
      "created_at": "2018-12-24T05:30:19+03:00",
      "updated_at": "2018-12-24T05:30:19+03:00"
  }
}
```


6.7.3 PUT - /api/v1/deals/{id}

Update a deal

```
PUT /api/v1/deals/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
title	required	Deal title
pipeline	required	Pipeline ID
stage_id	required	Deal stage ID
contact_person	required	Deal contact person
deal_value	optional	Deal value e.g 500
due_date	optional	The date a deal is expected to close
organization	optional	Deal Organization
status	optional	Deal status. Default open
source	optional	Deal source
currency	optional	Deal Currency

6.7.4 `DELETE` - /api/v1/deals/{id}

Delete a deal

```
DELETE /api/v1/deals/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.7.5 `GET` - /api/v1/deals

Get a list of all deals

```
GET /api/v1/deals
Accept: application/json
Authorization: Bearer $AccessToken
```

6.7.6 `GET` - /api/v1/deals/{id}/comments

Show deal comments

```
GET /api/v1/deals/{id}/comments
Accept: application/json
Authorization: Bearer $AccessToken
```

6.7.7 `GET` - /api/v1/deals/{id}/todos

Show deal todo list

```
GET /api/v1/deals/{id}/todos
Accept: application/json
Authorization: Bearer $AccessToken
```

6.7.8 GET - /api/v1/deals/{id}/calls

Show deal calls

```
GET /api/v1/deals/{id}/calls
Accept: application/json
Authorization: Bearer $AccessToken
```

6.7.9 GET - /api/v1/deals/{id}/products

Show deal products

```
GET /api/v1/deals/{id}/products
Accept: application/json
Authorization: Bearer $AccessToken
```

6.7.10 POST - /api/v1/deals/{id}/close

Close a deal

```
POST /api/v1/deals/{id}/close
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
id	required	Deal ID
lost_time	optional	Time when the deal was lost
status	required	Default lost
lost_reason	required	Reason why the deal was lost

6.8 Estimates

6.8.1 POST - /api/v1/estimates

Create a new estimate

```
POST /api/v1/estimates
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
client_id	required	Client ID
reference_no	required	Estimate Reference Number
due_date	required	Estimate due date
tax	required	Tax 1 percentage
tax2	required	Tax 2 percentage
discount	optional	Estimate discount percentage
currency	optional	Estimate Currency
deal_id	optional	Associated deal if any
is_visible	optional	Show/Hide estimate from client. Default 0
tags[]	optional	Array list of tags e.g `tags[design]`

6.8.2 GET - /api/v1/estimates/{id}

Get deal information

```
GET /api/v1/estimates/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response


```
{
  "type": "estimates",
  "id": "80",
  "attributes": {
      "id": 80,
      "reference_no": "EST-20181224-0080",
      "title": "Thomas Murazik Website Project",
      "client_id": 2,
      "deal_id": 0,
      "due_date": "2019-03-10T05:30:25+03:00",
      "tax": "2.43",
      "tax2": "0.00",
      "discount": "1.16",
      "discount_percent": 1,
      "currency": "USD",
      "notes": "Non dolorum placeat nisi at suscipit. Unde praesentium consequuntur␣
↪repellat doloribus voluptatem aliquam. Ut eum culpa consequatur cupiditate animi␣
↪rem. Eligendi consectetur pariatur recusandae neque inventore et alias.",
      "sent_at": null,
      "status": "Pending",
      "viewed_at": null,
      "invoiced_id": null,
      "invoiced_at": null,
      "accepted_time": null,
      "rejected_time": null,
      "rejected_reason": null,
      "exchange_rate": "1.00000",
      "sub_total": "0.00",
```


(continues on next page)

(continued from previous page)

```
        "amount": "0.00",
        "discounted": "0.00",
        "tax1_amount": "0.00",
        "tax2_amount": "0.00",
        "archived_at": null,
        "business": {
            "id": 2,
            "name": "Toy, Cormier and Kub",
            "contact_person": "muller.william@example.com"
        },
        "created_at": "2018-12-24T05:30:27+03:00",
        "updated_at": "2018-12-24T05:30:27+03:00"
    }
}
```

6.8.3 PUT - /api/v1/estimates/{id}

Update an estimate

```
PUT /api/v1/estimates/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
client_id	required	Client ID
reference_no	required	Estimate Reference Number
due_date	required	Estimate due date
tax	required	Tax 1 percentage
tax2	required	Tax 2 percentage
discount	optional	Estimate discount percentage
currency	optional	Estimate Currency
deal_id	optional	Associated deal if any
is_visible	optional	Show/Hide estimate from client. Default 0

6.8.4 DELETE - /api/v1/estimates/{id}

Delete an estimate

```
DELETE /api/v1/estimates/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.8.5 GET - /api/v1/estimates

Get a list of all estimates

```
GET /api/v1/estimates
Accept: application/json
Authorization: Bearer $AccessToken
```

6.8.6 GET - /api/v1/estimates/{id}/comments

Show estimate comments

```
GET /api/v1/estimates/{id}/comments
Accept: application/json
Authorization: Bearer $AccessToken
```

6.8.7 GET - /api/v1/estimates/{id}/items

Show estimate items

```
GET /api/v1/estimates/{id}/items
Accept: application/json
Authorization: Bearer $AccessToken
```

6.8.8 POST - /api/v1/estimates/{id}/invoice

Invoice an estimate

```
POST /api/v1/estimates/{id}/invoice
Accept: application/json
Authorization: Bearer $AccessToken
```

6.8.9 POST - /api/v1/estimates/{id}/copy

Duplicate the estimate

```
POST /api/v1/estimates/{id}/copy
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
id	required	Estimate ID

6.8.10 POST - /api/v1/estimates/{id}/send

Send estimate to client

```
POST /api/v1/estimates/{id}/send
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
id	required	Estimate ID
to[]	required	Comma separated list of email addresses
subject	required	Message subject

6.8.11 POST - /api/v1/estimates/{id}/project

Convert estimate to project

```
POST /api/v1/estimates/{id}/project
Accept: application/json
Authorization: Bearer $AccessToken
```

6.8.12 POST - /api/v1/estimates/{id}/cancel

Cancel estimate

```
POST /api/v1/estimates/{id}/cancel
Accept: application/json
Authorization: Bearer $AccessToken
```

6.9 Credits

6.9.1 POST - /api/v1/creditnotes

Create a new creditnote

```
POST /api/v1/creditnotes
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
client_id	required	Client ID
reference_no	required	Credit Reference Number
created_at	optional	Date created
tax	optional	Tax percentage
terms	optional	Creditnote terms
notes	optional	Creditnote notes
currency	optional	Creditnote Currency
is_refunded	optional	Set to 1 if credit was refunded
tags[]	optional	Array list of tags e.g `tags[design]`

6.9.2 `GET` - /api/v1/creditnotes/{id}

Get creditnote information

```
GET /api/v1/creditnotes/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response


```
{
  "type": "credits",
  "id": "10",
  "attributes": {
      "id": 10,
      "reference_no": "CN-20181224-0010",
      "client_id": 4,
      "status": "open",
      "currency": "USD",
      "tax": "1.85",
      "amount": "0.00",
      "balance": "0.00",
      "exchange_rate": "1.00000",
      "is_refunded": 0,
      "archived_at": null,
      "terms": null,
      "notes": null,
      "sent_at": null,
      "business": {
          "id": 4,
          "name": "Kertzmann, Schoen and Zboncak",
          "contact_person": "luna66@example.net"
      },
      "created_at": "2018-12-24T05:30:29+03:00",
      "updated_at": "2018-12-24T05:30:29+03:00"
  }
}
```

6.9.3 PUT - /api/v1/creditnotes/{id}

Update creditnote

```
PUT /api/v1/creditnotes/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
client_id	required	Client ID
reference_no	required	Credit Reference Number
created_at	optional	Date created
tax	optional	Tax percentage
terms	optional	Creditnote terms
notes	optional	Creditnote notes
currency	optional	Creditnote Currency
is_refunded	optional	Set to 1 if credit was refunded

6.9.4 DELETE - /api/v1/creditnotes/{id}

Delete an estimate

```
DELETE /api/v1/creditnotes/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.9.5 GET - /api/v1/creditnotes

Get a list of all creditnotes

```
GET /api/v1/creditnotes
Accept: application/json
Authorization: Bearer $AccessToken
```

6.9.6 GET - /api/v1/creditnotes/{id}/comments

Show creditnote comments

```
GET /api/v1/creditnotes/{id}/comments
Accept: application/json
Authorization: Bearer $AccessToken
```

6.9.7 GET - /api/v1/creditnotes/{id}/items

Show creditnote items

```
GET /api/v1/creditnotes/{id}/items
Accept: application/json
Authorization: Bearer $AccessToken
```

6.9.8 POST - /api/v1/creditnotes/{id}/use-credits

Use credits

```
POST /api/v1/creditnotes/{id}/use-credits
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
invoice_id	required	Invoice ID
creditnote_id	required	Credit Note ID
credited_amount	required	Amount to be credited e.g 50

6.9.9 POST - /api/v1/creditnotes/{id}/delete-credit

Use credits

```
POST /api/v1/creditnotes/{id}/delete-credit
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
id	required	Credit Note ID

6.9.10 POST - /api/v1/creditnotes/{id}/send

Send creditnote to client

```
POST /api/v1/creditnotes/{id}/send
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
id	required	Creditnote ID
to[]	required	Comma separated list of email addresses
subject	required	Email subject

6.10 Expenses

6.10.1 POST - /api/v1/expenses

Create a new expense

```
POST /api/v1/expenses
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
amount	required	Expense amount e.g 1500.00
category	required	Expense category
expense_date	required	Expense date
tax	required	Tax 1 percentage
tax2	required	Tax 2 percentage
currency	optional	Expense Currency
billable	optional	Whether the expense is billable. Default 1
notes	optional	Expense notes
project_id	optional	Associated project ID if any
client_id	optional	Associated client ID if any
vendor	optional	Associated vendor name
is_visible	optional	Show/Hide expense from client. Default 0
tags[]	optional	Array list of tags e.g `tags[design]`

6.10.2 GET - /api/v1/expenses/{id}

Get expense information

```
GET /api/v1/expenses/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response


```
{
  "type": "expenses",
  "id": "10",
  "attributes": {
      "id": 10,
      "code": "EXP-AC0010",
      "amount": "222.04",
      "before_tax": "0.00",
      "currency": "USD",
      "billable": 1,
      "category": 47,
      "vendor": "Feil and Sons",
```


(continues on next page)

(continued from previous page)


```
        "tax": "0.95",
        "tax2": null,
        "taxed": null,
        "expense_date": "2018-12-24T00:00:00+03:00",
        "billed": false,
        "project_id": 1,
        "client_id": 2,
        "invoiced_id": null,
        "is_recurring": 0,
        "frequency": null,
        "next_recur_date": null,
        "recur_starts": null,
        "recur_ends": null,
        "exchange_rate": "1.00000",
        "is_visible": 0,
        "notes": null,
        "user_id": 1,
        "created_at": "2018-12-24T05:30:44+03:00",
        "updated_at": "2018-12-24T05:30:44+03:00"
    }
}
```


6.10.3 `PUT` - /api/v1/expenses/{id}

Update an expense

```
PUT /api/v1/expenses/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
amount	required	Expense amount e.g 1500.00
category	required	Expense category
expense_date	required	Expense date
tax	required	Tax 1 percentage
tax2	required	Tax 2 percentage
currency	optional	Expense Currency
billable	optional	Whether the expense is billable. Default 1
notes	optional	Expense notes
project_id	optional	Associated project ID if any
client_id	optional	Associated client ID if any
vendor	optional	Associated vendor name
is_visible	optional	Show/Hide expense from client. Default 0
tags[]	optional	Array list of tags e.g `tags[design]`

6.10.4 `DELETE` - /api/v1/expenses/{id}

Delete an expense

```
DELETE /api/v1/expenses/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.10.5 GET - /api/v1/expenses

Get a list of all expenses

```
GET /api/v1/expenses
Accept: application/json
Authorization: Bearer $AccessToken
```

6.10.6 GET - /api/v1/expenses/{id}/comments

Show expense comments

```
GET /api/v1/expenses/{id}/comments
Accept: application/json
Authorization: Bearer $AccessToken
```

6.10.7 POST - /api/v1/expenses/{id}/copy

Duplicate expense

```
POST /api/v1/expenses/{id}/copy
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
id	required	Expense ID

6.11 Payments

6.11.1 POST - /api/v1/payments

Create a new payment

```
POST /api/v1/payments
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
invoice_id	required	Invoice ID
payment_date	required	Date when the payment was made
amount	required	Amount of payment made
payment_method	required	Payment method ID
gateway	required	`Must be set to offline`
notes	optional	Payment additional notes
currency	optional	Payment Currency
send_email	optional	If an email should be sent to client. Default 1

6.11.2 `GET` - /api/v1/payments/{id}

Get payment information

```
GET /api/v1/payments/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response


```
{
  "type": "payments",
  "id": "10",
  "attributes": {
      "id": 10,
      "code": "PAY-20181224-0010",
      "invoice_id": 7,
      "payment_method": "Cash",
      "curreny": null,
      "amount": "73.68",
      "notes": null,
      "payment_date": "2018-12-19T05:30:31+03:00",
      "exchange_rate": "1.00000",
      "project_id": null,
      "refunded": 0,
      "archived_at": null,
      "business": {
          "id": 5,
          "name": "Rogahn-Gerhold",
          "contact_person": "wiza.samanta@example.org"
      },
      "created_at": "2018-12-24T05:30:31+03:00",
      "updated_at": "2018-12-24T05:30:31+03:00"
  }
}
```


6.11.3 `PUT` - /api/v1/payments/{id}

Update a payment

```
PUT /api/v1/payments/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
invoice_id	required	Invoice ID
payment_date	required	Date when the payment was made
amount	required	Amount of payment made
payment_method	required	Payment method ID
notes	optional	Payment additional notes
currency	optional	Payment Currency

6.11.4 DELETE - /api/v1/payments/{id}

Delete a payment

```
DELETE /api/v1/payments/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.11.5 GET - /api/v1/payments

Get a list of all payments

```
GET /api/v1/payments
Accept: application/json
Authorization: Bearer $AccessToken
```

6.11.6 GET - /api/v1/payments/{id}/comments

Show estimate comments

```
GET /api/v1/payments/{id}/comments
Accept: application/json
Authorization: Bearer $AccessToken
```

6.11.7 POST - /api/v1/payments/{id}/refund

Mark a payment as refunded

```
POST /api/v1/payments/{id}/refund
Accept: application/json
Authorization: Bearer $AccessToken
```

6.12 Contracts

6.12.1 POST - /api/v1/contracts

Create a new contracts

```
POST /api/v1/contracts
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
contract_title	required	Contract title
client_id	required	Client associated with the contract
start_date	required	Contract start date
end_date	required	Contract end date
expiry_date	required	Number of days before a contract expires. E.g 14
payment_terms	optional	Number of days. E.g 14
currency	optional	Contract Currency
termination_notice	optional	Number of days to be notified before termination
rate_is_fixed	optional	If fixed rate. Default 0
fixed_rate	optional	Contract fixed amount e.g 1500.00
hourly_rate	optional	Contract hourly rate
description	optional	Contract description
license_owner	optional	Contract license owner. `freelancer or client`
late_payment_fee	optional	Late payment fee
late_fee_percent	optional	If late payment is percentage. Default 1
cancelation_fee	optional	Contract cancellation fee
is_visible	optional	Show/hide contract from client
deposit_required	optional	Amount of deposit required. E.g 1500.00
services	optional	List of contract services. E.g Web Design, SEO
client_rights	optional	Rights granted to client
portfolio_rights	optional	Right to include work in portfolio. Default 1
non_compete	optional	Add non-compete section. Default 1
appropriate_conduct	optional	Enable sexual harassment clause. Default 1

6.12.2 GET - /api/v1/contracts/{id}

Get contract information

```
GET /api/v1/contracts/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response

```
{
  "type": "contracts",
  "id": "10",
  "attributes": {
      "id": 10,
      "title": "Dare LLC Contract",
      "start_date": "2018-12-24T05:30:20+03:00",
      "end_date": "2019-01-08T05:30:20+03:00",
      "expiry_date": "2018-12-29T05:30:20+03:00",
      "rate_is_fixed": 0,
      "fixed_rate": null,
      "hourly_rate": "12.97",
      "currency": "USD",
      "license_owner": "client",
      "payment_terms": "6",
      "late_payment_fee": "0.00",
      "late_fee_percent": 1,
      "termination_notice": 12,
      "cancelation_fee": "11.19",
      "deposit_required": "0.00",
      "signed": 0,
      "services": "Beatae blanditiis ea commodi et tempore est.",
      "client_rights": "Qui culpa qui consequatur architecto nam officia. Minus nulla␣
↪odio sapiente delectus ut. Dolore nemo reprehenderit dolore odit eum consequuntur.␣
↪Voluptate nesciunt et vero beatae sint ut.",
      "portfolio_rights": 1,
      "non_compete": 1,
      "feedbacks": 0,
      "appropriate_conduct": 1,
      "annotations": null,
      "description": "Necessitatibus totam qui nostrum ad non qui distinctio. Ipsam␣
↪non sed deserunt recusandae non eum amet. Et quo quaerat enim voluptates pariatur.␣
↪Dolor sint cum voluptatem enim. Et ratione deleniti aut deserunt eligendi itaque␣
↪aut. Qui et eius non voluptatibus quos a sunt. Et rerum quia suscipit nisi.␣
↪Voluptatem exercitationem culpa at quo deleniti. Reprehenderit repellat ullam nemo␣
↪tempore amet optio. Et porro distinctio nostrum minus placeat. Voluptatum dolore ex␣
↪in qui esse occaecati eum. Minus iste nostrum id laudantium. Vel ipsam qui expedita␣
↪sed et laborum commodi unde. Temporibus fugit sint voluptas fuga.",
      "viewed_at": null,
      "sent_at": null,
      "is_draft": true,
      "rejected_at": null,
      "rejected_reason": null,
      "user_id": 1,
      "business": {
          "id": 1,
          "name": "Sipes-Schuster",
          "contact_person": "ehauck@example.com"
      },
      "created_at": "2018-12-24T05:30:20+03:00",
      "updated_at": "2018-12-24T05:30:20+03:00"
  }
}
```

6.12.3 PUT - /api/v1/contracts/{id}

Update a contract

```
PUT /api/v1/contracts/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Tip: Same as the create new contract parameters

6.12.4 DELETE - /api/v1/contracts/{id}

Delete a contract

```
DELETE /api/v1/contracts/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.13 Clients

6.13.1 POST - /api/v1/clients

Create a new client

```
POST /api/v1/clients
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
name	required	Client Name
email	required	Client email address
contact_email	required	Contact email address
phone	optional	Client phone number
address1	optional	Address
zip_code	optional	Zip Code
city	optional	City
state	optional	State
locale	optional	Preferred locale
country	optional	Country
tax_number	optional	Client tax number if any
currency	optional	Preferred currency
website	required	Client website URL
facebook	required	Client facebook link
twitter	optional	Twitter account URL
skype	optional	Skype address
linkedin	optional	LinkedIn profile
notes	optional	Additional notes
tags[] | optional | Array list of tags e.g `tags[design]`		

6.13.2 `GET` - /api/v1/clients/{id}

Get client information

```
GET /api/v1/clients/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response


```
{
  "type": "clients",
  "id": "100",
  "attributes": {
      "id": 100,
      "name": "Greenholt-Harris",
      "code": "COM00100",
      "email": "mclaughlin.jason@example.net",
      "contact": {
          "id": 1,
          "email": "admin@example.com",
          "name": "William Mandai"
      },
      "address": {
          "address1": "402 Reynolds Trace\nNorth Lutherchester, SD 94456-5868",
          "address2": null,
          "city": "East Geo",
          "state": null,
```


(continues on next page)

(continued from previous page)


```
            "zipcode": null,
            "country": "Peru"
        },
        "website": "https://hartmann.com",
        "phone": null,
        "mobile": null,
        "tax_number": null,
        "currency": "USD",
        "expense": "0.00",
        "balance": "0.00",
        "paid": "0.00",
        "social": {
            "skype": null,
            "facebook": null,
            "twitter": null,
            "linkedin": null
        },
        "notes": "Neque veritatis pariatur ut voluptatum. Qui officia molestias␣
↪distinctio dicta quibusdam. Amet et adipisci ad eveniet.",
        "logo": "/storage/logos/tux_droid_1.jpg",
        "unsubscribed_at": null,
        "created_at": "2018-12-24T05:30:17+03:00",
        "updated_at": "2018-12-24T05:30:17+03:00"
    }
}
```


6.13.3 PUT - /api/v1/clients/{id}

Update client information

```
PUT /api/v1/clients/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Tip: Same as the create new client API parameters

6.13.4 DELETE - /api/v1/clients/{id}

Delete a client

```
DELETE /api/v1/clients/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.13.5 GET - /api/v1/clients

Get a list of all clients

```
GET /api/v1/clients
Accept: application/json
Authorization: Bearer $AccessToken
```

6.13.6 GET - /api/v1/clients/{id}/contacts

Show client contacts

```
GET /api/v1/clients/{id}/contacts
Accept: application/json
Authorization: Bearer $AccessToken
```

6.13.7 GET - /api/v1/clients/{id}/projects

Show client projects

```
GET /api/v1/clients/{id}/projects
Accept: application/json
Authorization: Bearer $AccessToken
```

6.13.8 GET - /api/v1/clients/{id}/invoices

Show client invoices

```
GET /api/v1/clients/{id}/invoices
Accept: application/json
Authorization: Bearer $AccessToken
```

6.13.9 GET - /api/v1/clients/{id}/estimates

Show client estimates

```
GET /api/v1/clients/{id}/estimates
Accept: application/json
Authorization: Bearer $AccessToken
```

6.13.10 GET - /api/v1/clients/{id}/payments

Show client payments

```
GET /api/v1/clients/{id}/payments
Accept: application/json
Authorization: Bearer $AccessToken
```

6.13.11 GET - /api/v1/clients/{id}/subscriptions

Show client subscriptions

```
GET /api/v1/clients/{id}/subscriptions
Accept: application/json
Authorization: Bearer $AccessToken
```

6.13.12 GET - /api/v1/clients/{id}/expenses

Show client expenses

```
GET /api/v1/clients/{id}/expenses
Accept: application/json
Authorization: Bearer $AccessToken
```

6.13.13 GET - /api/v1/clients/{id}/deals

Show organization deals

```
GET /api/v1/clients/{id}/deals
Accept: application/json
Authorization: Bearer $AccessToken
```

6.14 Contacts

6.14.1 POST - /api/v1/contacts

Create a new contact

```
POST /api/v1/contacts
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
name	required	Contact Name
email	required	Contact email address
username	required	Contact username
company	optional	Contact Company ID
password	optional	Contact login password
phone	optional	Contact Phone Number
invite	optional	Send email invitation. Set to 1 to send email

6.14.2 GET - /api/v1/contacts/{id}

Get contact information

```
GET /api/v1/contacts/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response


```
{
  "type": "contacts",
  "id": "10",
  "attributes": {
      "id": 10,
      "name": "Johnathan Yundt I",
      "job_title": "Floral Designer",
      "email": "mackenzie46@example.org",
      "avatar": "/storage/avatars/avatar9.png",
      "city": null,
      "country": null,
      "website": null,
      "hourly_rate": "17.00",
      "business": {
          "id": 6,
          "name": "Turcotte, Buckridge and Herman",
          "contact_person": "luna66@example.net",
          "currency": "USD",
          "balance": "0.00",
          "expense": "0.00",
          "paid": "0.00"
      },
      "created_at": "2018-12-24T05:30:09+03:00",
      "updated_at": "2018-12-24T05:30:16+03:00"
  }
}
```


6.14.3 PUT - /api/v1/contacts/{id}

Update contact information

```
PUT /api/v1/contacts/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Tip: Same as the create contact API parameters

6.14.4 DELETE - /api/v1/contacts/{id}

Delete a contact

```
DELETE /api/v1/contacts/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.14.5 GET - /api/v1/contacts

Get a list of all contacts

```
GET /api/v1/contacts
Accept: application/json
Authorization: Bearer $AccessToken
```

6.15 Projects

6.15.1 POST - /api/v1/projects

Create a new projects

```
POST /api/v1/projects
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
name	required	Project Name
client_id	required	Project client ID
start_date	required	Project start date
due_date	required	Project due date
currency	optional	Project Currency
description	optional	Description
hourly_rate	optional	Hourly rate
fixed_price	optional	Fixed Price. E.g 3400.00
notes	optional	Project Notes
manager	optional	User ID
estimate_hours	optional	Project Estimated hours
billing_method	optional	`hourly_staff_rate, hourly_task_rate, hourly_project_rate, fixed_rate`
tags[]	optional	Array list of tags e.g `tags[design]`

6.15.2 GET - /api/v1/projects/{id}

Get project information

```
GET /api/v1/projects/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response

```
{
  "type": "projects",
  "id": "6",
  "attributes": {
      "id": 6,
      "name": "Rice, Doyle and Bauch Project",
      "code": "PRO0006",
      "description": "Earum quia quis qui id minima et. Esse facere qui eligendi et eaque quia. Rerum corporis consequatur velit odit quam. Aliquam quia architecto et et repellendus. Molestiae et facilis neque dolor. Et laudantium totam aut et. Recusandae corrupti non maxime sed ratione eos ut. Cupiditate repellat harum quia dolor. Et voluptatum laboriosam ex nostrum sed necessitatibus repellat. Eveniet sunt enim est aut ea minima eos. Culpa nihil rem qui non sunt quia. Sed et adipisci porro dolore perferendis fugiat. Quisquam laboriosam quisquam et aspernatur. Rem vel ad facere enim cumque.",
      "client_id": 3,
      "business": {
          "id": 3,
          "name": "Ferry-Schuster",
          "contact_person": "wiza.samanta@example.org"
      },
      "currency": "USD",
      "start_date": "2018-12-24T00:00:00+03:00",
      "due_date": "2019-03-01T00:00:00+03:00",
      "hourly_rate": "59.38",
      "fixed_price": "0.00",
      "progress": 0,
      "notes": null,
      "manager": 1,
      "status": "Active",
      "estimate_hours": "82.08",
      "used_budget": "0.00",
      "billable_time": "0.00",
      "unbillable_time": "0.00",
      "unbilled": "0.00",
      "sub_total": "0.00",
      "total_expenses": "0.00",
      "contract_id": null,
      "billing_method": "hourly_project_rate",
      "created_at": "2018-12-24T05:30:32+03:00",
      "updated_at": "2018-12-24T05:30:32+03:00"
  }
}
```

6.15.3 PUT - /api/v1/projects/{id}

Update project information

```
PUT /api/v1/projects/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Tip: Same as the create new project API parameters

6.15.4 DELETE - /api/v1/projects/{id}

Delete project

```
DELETE /api/v1/projects/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.15.5 GET - /api/v1/projects

Get a list of all projects

```
GET /api/v1/projects
Accept: application/json
Authorization: Bearer $AccessToken
```

6.15.6 GET - /api/v1/projects/{id}/invoices

Show project invoices

```
GET /api/v1/projects/{id}/invoices
Accept: application/json
Authorization: Bearer $AccessToken
```

6.15.7 GET - /api/v1/projects/{id}/tasks

Show project tasks

```
GET /api/v1/projects/{id}/tasks
Accept: application/json
Authorization: Bearer $AccessToken
```

6.15.8 GET - /api/v1/projects/{id}/expenses

Show project expenses

```
GET /api/v1/projects/{id}/expenses
Accept: application/json
Authorization: Bearer $AccessToken
```

6.15.9 POST - /api/v1/projects/{id}/done

Mark project as done

```
POST /api/v1/projects/{id}/done
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
id | required | Project ID		

6.15.10 POST - /api/v1/projects/{id}/invoice

Invoice project

```
POST /api/v1/projects/{id}/invoice
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
invoice_style	required	`single or task_line`
expense[]	optional	Array list of expense IDs to include

6.15.11 POST - /api/v1/projects/{id}/copy

Duplicate a project

```
POST /api/v1/projects/{id}/copy
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
id	required	Project ID
parts[]	optional	Array list of what to clone e.g `parts[expenses], parts[tasks]`

6.16 Tickets

6.16.1 POST - /api/v1/tickets

Create a new ticket

```
POST /api/v1/tickets
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
department	required	Ticket department ID
subject	required	Ticket subject
body	required	Ticket Message
project_id	optional	Project ID associated with the ticket

6.16.2 GET - /api/v1/tickets/{id}

Get ticket information

```
GET /api/v1/tickets/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response


```
{
  "type": "tickets",
  "id": "10",
  "attributes": {
      "id": 10,
      "subject": "Arvid Ticket",
      "code": "TKT-20181224-0010",
      "body": "Iure et laborum debitis quod veniam eum vel temporibus. Et id culpa␣
↪asperiores molestiae qui animi ad necessitatibus. Ea unde corporis omnis. Minus est␣
↪dignissimos cupiditate facere autem. Quia natus aliquam qui et. Incidunt et␣
↪deleniti tempore ut repellat accusamus sed. Hic quasi dolores minima molestiae.␣
↪Sint non cumque repellat alias vero et perspiciatis. Ad enim qui rerum libero.␣
↪Labore aut voluptas dolores possimus tenetur. Vero maxime facilis aut debitis est␣
↪quis dignissimos. Quae ipsa id nihil illo. In omnis ratione sunt quo est et officia.
↪ In repudiandae recusandae ipsa similique beatae adipisci.",
      "status": {
          "id": 1,
          "name": "open"
      },
      "department": {
          "id": 1,
          "name": "Billing"
```


(continues on next page)

(continued from previous page)

```
        },
        "user_id": 1,
        "project_id": null,
        "priority": {
            "id": 1,
            "name": "Low"
        },
        "due_date": "2018-12-27T00:00:00+03:00",
        "closed_at": null,
        "assignee": {
            "id": 1,
            "name": "William Mandai"
        },
        "resolution_time": 0,
        "archived_at": null,
        "created_at": "2018-12-24T05:30:44+03:00",
        "updated_at": "2018-12-24T05:30:44+03:00"
    }
}
```

6.16.3 PUT - /api/v1/tickets/{id}

Update ticket information

```
PUT /api/v1/tickets/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Tip: Same as the create new ticket API parameters above

6.16.4 DELETE - /api/v1/tickets/{id}

Delete ticket

```
DELETE /api/v1/tickets/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.16.5 GET - /api/v1/tickets

Get a list of all tickets

```
GET /api/v1/tickets
Accept: application/json
Authorization: Bearer $AccessToken
```

6.16.6 GET - /api/v1/tickets/{id}/comments

Show ticket comments

```
GET /api/v1/tickets/{id}/comments
Accept: application/json
Authorization: Bearer $AccessToken
```

6.16.7 POST - /api/v1/tickets/{id}/status

Update ticket status

```
POST /api/v1/tickets/{id}/status
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
status	required	Ticket status ID

6.17 Tasks

6.17.1 POST - /api/v1/tasks

Create a new task

```
POST /api/v1/tasks
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
project_id	required	Project ID
user_id	required	Task creator user id
name	required	Task Name
start_date	optional	Task start date
due_date	optional	Task due date
hourly_rate	optional	Hourly rate e.g 30.00
milestone_id	optional	Milestone ID
stage_id	optional	Task stage ID
team[]	optional	Array list of team member ID's
estimated_hours	optional	Task estimated hours e.g 72
description	optional	Description
visible	optional	Hide or show to client
tags[]	optional	Array list of tags e.g `tags[design]`

6.17.2 GET - /api/v1/tasks/{id}

Get task information

```
GET /api/v1/tasks/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response


```
{
  "type": "tasks",
  "id": "10",
  "attributes": {
      "id": 10,
      "name": "McGlynn-Jaskolski Task",
      "project": {
          "id": 1,
          "name": "Stracke PLC Project"
      },
      "milestone": {
          "id": null,
          "name": null
      },
      "progress": 94,
      "hourly_rate": "4.08",
      "estimated_hours": "0.00",
      "estimated_price": "$0.00",
      "hours": 0,
      "start_date": "2019-01-07T00:00:00+03:00",
      "due_date": "2019-01-16T00:00:00+03:00",
      "description": "Vel autem ea aperiam nihil. Consequatur neque omnis omnis ut␣
↪fugiat amet dolores. Voluptates quisquam odit tenetur doloremque ipsa voluptates.",
      "updated_at": "2018-12-24T05:30:32+03:00",
      "created_at": "2018-12-24T05:30:32+03:00"
  }
}
```


6.17.3 PUT - /api/v1/tasks/{id}

Update task information

```
PUT /api/v1/tasks/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Tip: Same as the create task API parameters above

6.17.4 DELETE - /api/v1/tasks/{id}

Delete a task

```
DELETE /api/v1/tasks/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.17.5 GET - /api/v1/tasks

Get a list of all tasks

```
GET /api/v1/tasks
Accept: application/json
Authorization: Bearer $AccessToken
```

6.17.6 POST - /api/v1/tasks/{id}/copy

Duplicate a task

```
POST /api/v1/tasks/{id}/copy
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
project_id	required	Project ID to copy task to

6.18 Todos

6.18.1 POST - /api/v1/todos

Create a new todo

```
POST /api/v1/todos
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name | Required | Description		
module	required	Module related to todo e.g `deals, clients, leads`
module_id	required	Entity ID e.g `18`
subject	required	Todo subject
due_date	optional	Todo start date
assignee	optional	User ID of the person responsible
notes	optional	Additional notes

6.18.2 `GET` - /api/v1/todos/{id}

Get todo information

```
GET /api/v1/todos/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Sample Response


```
{
  "type": "todos",
  "id": "1",
  "attributes": {
      "id": 1,
      "subject": "Send proposal ",
      "order": 0,
      "parent": 0,
      "due_date": "2019-01-07T00:00:00+03:00",
      "notes": null,
      "assignee": {
          "id": 1,
          "name": "William Mandai"
      },
      "reminded_at": null,
      "entity": {
          "id": 35,
          "name": "VonRueden PLC Deal"
      },
      "is_visible": 0,
      "completed": 0,
      "user_id": 1,
      "created_at": "2019-01-06T20:00:36+03:00",
      "updated_at": "2019-01-06T20:00:36+03:00"
  }
}
```


6.18.3 `PUT` - /api/v1/todos/{id}

Update todo information

```
PUT /api/v1/todos/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

Parameters

Name	Required	Description
subject	required	Todo subject
due_date	optional	Todo start date
assignee	optional	User ID of the person responsible
notes	optional	Additional notes

6.18.4 `DELETE` - /api/v1/todos/{id}

Delete a todo

```
DELETE /api/v1/todos/{id}
Accept: application/json
Authorization: Bearer $AccessToken
```

6.18.5 `GET` - /api/v1/todos

Get a list of all todos

```
GET /api/v1/todos
Accept: application/json
Authorization: Bearer $AccessToken
```

CHAPTER 7

Migration

Note: Unlike freelancer office, Workice CRM now requires PHP >= 7.2 and MySQL.

To purchase Workice with a discounted price for Users with Freelancer Office License, visit https://desk.workice.com and register. After Registration, login and click **Buy Workice**. You can download the new Workice CRM source code from your support page https://desk.workice.com (Profile > Downloads) section and proceed with Installation.

Attention: The discounted price is only available when paying via PayPal. We strongly recommend buyers to thoroughly check the item description, comments and demo (https://app.workice.com) to see if it works for your situation. If you have a pre-sale question please ask before purchase we will only offer refund incase we can't fix the issue.

Attention: Do not replace freelancer office files with Workice files install Workice separately before importing data.

Attention: Importing data from freelancer office to Workice requires CRON to be already running see the steps on how to setup cron here

7.1 Export data as JSON

Download the import file from here

Extract the downloaded file and upload it to your Root folder where **freelancer office** is installed.

- Access the URL where you uploaded the above file example **http://your-freelancer-office.com/workice-importer.php**.

- The import file will attempt to generate a JSON version of your data and create a file named **crmdata.json** in Root folder (We will require that file to import the data to Workice CRM).

> **Attention:** Download the JSON file (crmdata.json) from your server and delete file **workice-importer.php** and **crmdata.json**

7.2 Import to Workice

- Login to Workice CRM as administrator.
- Go to **Settings** > **System Info** > **Import** > **Freelancer Office**.
- When a pop modal appears, choose the file **crmdata.json** (that you downloaded from your server as described above).
- Click on **Save** button to start importing data.

Note: Workice will send an email to you when import data has been completed successfully

If your data does not show correctly after importing, run the command below or click on the command button in **Settings > Info > Commands button > app:balances** button;

```
php artisan app:balances
```

This command will attempt to re-calculate your invoices, estimates etc

7.3 Need help?

You can contact us here for help or send us an email to support@workice.com

CHAPTER 8

Introduction

Workice CRM is a place for you to store information on the people and clients you interact with and track your outreach efforts. The CRM includes sales and marketing dashboards, invoicing dashboard, expenses dashboard, payments dashboard, projects and ticketing dashboards.

Note: Workice CRM is build with Laravel Framework

To purchase Workice CRM Click Here

8.1 Top Navigation

In your Workice account, click your login name next to your avatar in the top navigation bar. The following sections will appear in the dropdown menu:

- Settings - Your Profile settings including GDPR.
- Tell a friend - Like Workice CRM? Share it with a friend.
- Reminders - Shows a list of all your reminders.
- Notifications - A list of all your notifications.
- Canned Responses - Instead of repeatedly copying and pasting the same email response you can save time by using canned responses.
- Need Help? - Have an issue? Open a ticket with us directly within your Workice CRM.

8.2 Invoicing Dashboard

This is the first page you'll see when you login. It provides a general overview of the invoicing status of your freelance business. The page displays the invoice-specific data that's associated with your business. It also allows you to see the updates at a glance.

The dashboard is designed to offer a simple yet powerful overview of your total business accounts:

- **Outstanding**: The total amount of all unpaid invoices.
- **Invoiced**: The total amount invoiced so far.
- **This Month**: Total amount invoiced in the current month.
- **Last Month**: Total amount of invoices created last month.

Tip: If you are being paid in a range of currencies, the dashboard will display the totals in your Base Currency.

Chart Below the stats boxes, you'll see a chart presenting your invoicing data in an easy-to-understand graphical format. The data presented in the chart are total invoiced amount versus revenue collected in the current Year.

8.3 Sales Dashboard

Sales dashboard provides a general overview of deals and leads.

The dashboard provides the information below:

- **Won Deals**: Total amount of won deals.
- **Lost Deals**: The total amount of lost deals.
- **This Month**: Total amount of deals this month.
- **Forecasted**: Get an idea of the revenue projection for your company.

Tip: If you are being paid in a range of currencies, the dashboard will display the totals in your Base Currency.

Chart Below the stats boxes, you'll see a chart presenting your sales data in an easy-to-understand graphical format. The data presented in the chart are total won deals versus lost deals in the current Year.

8.4 Expenses Dashboard

Expenses dashboard provides a general overview of your expenses.

The dashboard provides the information below:

- **Billed**: Total amount of billed expenses.
- **Unbillable**: Total amount of unbillable expenses.
- **This Month**: Total amount of expenses this month.
- **Last Month**: Total amount of expenses last month.

Tip: If you are being paid in a range of currencies, the dashboard will display the totals in your Base Currency.

Chart Below the stats boxes, you'll see a chart presenting your expenses data in an easy-to-understand graphical format. The data presented in the chart are total amount of expenses billable versus total amount billed in the current Year.

8.5 Payments Dashboard

Payment dashboard provides a general overview of your payments status of your freelance business.

The dashboard provides the information below:

- **Receipts**: Total amount received so far.
- **Credits**: Total amount of creditnotes issued.
- **This Month**: Total amount of payments received this month.
- **Last Month**: Total amount of payments received last month.

Tip: If you are being paid in a range of currencies, the dashboard will display the totals in your Base Currency.

Chart Below the stats boxes, you'll see a chart presenting your payments data in an easy-to-understand graphical format. The data presented in the chart are total amount of payments received versus total amount creditnotes issued in the current Year.

8.6 Projects Dashboard

Project dashboard provides a general overview of your project status of your freelance business.

The dashboard provides the information below:

- **Active**: Total number of active projects.
- **Done**: Total number of completed projects.
- **Pending Tasks**: Total number of pending tasks.
- **Done Tasks**: Total number of completed tasks.

Chart Below the stats boxes, you'll see a chart presenting your projects data in an easy-to-understand graphical format. The data presented in the chart are total number of projects, tasks and issues in the current Year.

8.7 Ticketing Dashboard

Ticketing dashboard provides a general overview of your support status for your freelance business.

The dashboard provides the information below:

- **Open**: Total number of open tickets.
- **Closed**: Total number of closed tickets.
- **Tickets**: Total number tickets.
- **Response Time**: Average amount of time used to solve a ticket

Chart Below the stats boxes, you'll see a chart presenting your tickets data in an easy-to-understand graphical format. The data presented in the chart are total number of tickets versus the number of ticket replies in the current Year.

8.8 Right Sidebar: Activities

The Activity list is incredibly useful as it presents an up-to-date, action-packed summary of what is happening across your entire CRM. Every action taken, whether by you or by one of your clients, is listed in chronological order, together with the date the action occurred.

CHAPTER 9

Deals

Clicking on the "Deals" tab in Workice CRM takes you to our comprehensive CRM, a feature exclusive to our Small Business and Enterprise plans.

Tip: You can choose the deal view type ie. **Kanban view** or **Table View** by selecting the Table icon on the deals page.

9.1 What are deals?

Think of deals as a way of identifying qualified leads as sales opportunities. Deals are contacts that no longer sit dormant in your database; they're actually in your pipeline and are on track to become customers. When a deal is created it is assigned to an internal team member who monitors and manages the movement of deals from stage to stage in your pipelines.

9.2 What are pipelines?

Pipelines are your business processes — sales-related and otherwise. For example, sales pipelines could be "Orders" or "Sales" They represent the entire process of converting qualified leads into customers.

You can setup as many pipelines (or processes) as necessary for your business since Workice CRM does not limit the amount of active pipelines you can create.

Tip: To add pipelines, go to **Settings** - **Deal Settings** and click on **Sales Pipelines** button.

9.3 What are stages?

Pipelines contain stages. If pipelines are your processes, stages are the steps that deals must advance through to complete those processes (and in many cases, for you to close sales).

For example, stages could be "Need To Contact" and "Currently In Contact" and "Send Proposal".

Tip: Click the Settings icon in the upper right corner of the deals page to add more stages.

Open deals are those that are active. **Won** deals are those that achieve the goal of the pipeline (e.g. buy a product or are successfully on-boarded). **Lost** deals are leads that enter your pipelines but do not convert.

9.4 Deals Forecasting

To know how much money your business has made is useful, but sometimes you need to know how much money you could be making in the future.In Workice CRM, we build deal forecast feature to help you get an idea of the revenue projection for your freelance business. Click on the ellipsis (3 dots) on the deal page to access **Deal Forecast**.

You can view won, lost and archived deals by clicking on the ellipsis button and filter your deals.

9.5 Won or Lost Deal

Tip: To close a deal, click on the green won button. If a deal is lost, click on the red lost button and specify a reason why the deal was lost (optional).

9.6 Log a call

To log a deal call record: Open a deal and click on **Calls** tab. Click on the Phone icon on the right. A modal will open for you to enter record details;

- `Subject` - Call subject e.g Product Demo
- `Call Type` - Either inbound or outbound
- `Duration` - The amount of time spent on the call in format 00:30:00. Example; 00:10:00 means 10minutes
- `Assignee` - The person responsible.
- `Schedule call` - You can schedule a phone call and the CRM will remind you.
- `Description` - Call description
- `Call Result` - Summary of what you discussed on your call conversation.

9.7 Emails

To send email to a deal, click on a deal and select the Emails tab. Whether you are following up a deal or you received an email reply from a deal, all the emails will be listed on the deal emails page.

Attention: To display emails replied by your deal on the CRM, you need to setup email integration in **Settings** - **Deal Settings** section

9.8 Activities

You can add deal activities e.g "Followup call" to a deal using the Activity tab.

Tip: To enter a todo task quickly, you may use english words for dates inside square brackets e.g `Send proposal [tomorrow]` or `Followup on deal acme [next monday]`.

9.9 Products

Add deal products using the Products tab. When you add a product the deal value will be updated with the product price.

9.10 Files

Use the files tab to add files associated with the deal.

9.11 Comments

Need to discuss something with the sales person attached to the deal? Use the deal comments section.

9.12 Reminders

Use the reminder icon on the top menu of the deal view to set reminders.

CHAPTER 10

Leads

Lead management is an important part of any sales process. When you get information about someone who might be interested in your products or services, but haven't yet verified their interest or fit for your business, you'll add them to Workice CRM on the Leads section.

Once you've determined a lead is qualified, you'll convert the lead and Workice will create a sales opportunity(deal), a contact, and an organization for you.

Leads could be obtained through a purchased contact list, a form on your website, trade shows and meetings where you've gathered business cards, and other marketing efforts. You can create leads in a few different ways:

- Enter them on the Leads section manually.
- Import them from a CSV or Excel file.
- Email integrations. Create new leads whenever you receive a new email.
- Use zappier to scan business cards using https://www.fullcontact.com/ and send POST data to your Workice CRM.
- Add a Web to Lead form to your website so visitors can submit their own information.
- If you have technical resources, you can also add leads through Workice CRM API.

Tip: You can choose how to view leads ie. **Kanban view** or **Table View** by selecting the Table icon on the leads page.

Tip: To add lead stages, go to **Settings - Lead Settings** and click on **Stages** button.

Tip: Mass email the leads by selecting the check boxes corresponding to the records and clicking on the Send Mail button. `You will need to view the leads as a table`

10.1 Adding leads manually

To create a lead directly in Workice CRM:

Click on **create** button on the leads section.
Enter the lead's information. (Fields marked in a * are required).
Click Save button to add the lead.

Tip: You can move leads to next stage by clicking on **Next Stage** while in Lead View page.

10.2 Log a call

To log a lead call record: Open a lead and click on **Calls** tab. Click on the Phone icon on the right. A modal will open for you to enter record details;

- `Subject` - Call subject e.g Product Demo
- `Call Type` - Either inbound or outbound
- `Duration` - The amount of time spent on the call in format 00:30:00. Example; 00:10:00 means 10minutes
- `Assignee` - The person responsible.
- `Schedule call` - You can schedule a phone call and get email alerts to remind you.
- `Description` - Call description
- `Call Result` - Summary of what you discussed on your call conversation.

10.3 Emails

To send email to a lead, click on a lead and select the Emails tab. Whether you are following up on a lead or you received an email reply from a lead, all the emails will be listed on the emails page.

Attention: To display emails replied by your lead on the CRM, you need to setup email integration in **Settings - Lead Settings** section

10.4 Activities

You can add lead activities e.g "Followup call" to a lead using the Activity tab.

Tip: To enter a todo task quickly, you may use english words for dates inside square brackets e.g `Send proposal [tomorrow]` or `Followup on lead John Doe [next friday]`.

10.5 Files

Use the files tab to add files associated with the lead.

10.6 Comments

Need to discuss something with the sales person attached to the lead? Use the comments section.

10.7 Reminders

Use the reminder icon on the top menu of the lead view to set reminders.

10.8 Convert Lead

Once the lead status has reached a certain stage, it can be qualified as a potential/deal. - After converting a lead into a potential/deal, lead status will be changed to Converted. - You can view the list of converted leads by selecting the Converted Leads list view. - When a lead is converted, it cannot be reverted to a lead. - An account will be created on lead conversion if the Company Name detail is available for the lead.

10.9 GDRP - Double opt in

Double opt-in offers an extra confirmation step to verify each lead email address added to your CRM database. When this setting is enabled, leads who submit a form receive a follow-up email with a request for consent.

Tip: To enabled double opt in in leads, go to **Settings** - **Lead Settings** section.

CHAPTER 11

Clients

Your clients are the core of your freelance business. Clients in WorkiceCRM will typically hold all information specific to a company that your organisation will have a relationship with. In real world terms an Account may be a business entity that is a qualified Sales Prospect, Customer, Supplier or Re-seller and can be used to track all interactions that take place between these entities and your organisation. The relationship between the account and contact records is one-to-many, such that there can be many contacts associated with a single account.

11.1 Contacts

In Workice, a Contact is an individual who is typically associated with a client (organisation) or Opportunity (qualified prospect). For example if Apple is the Client, then John Smith, Sales Manager of Apple is the Contact. This module holds all information relating to these individuals and also provides a vantage point for any history relating to a Contact record, for example if they sent you an Email.

11.2 List Clients

The Clients page is a list page that presents a summary of all your clients in a user-friendly table. Think of Accounts page as the "central station" of your client activity. Most of your day-to-day invoicing actions can be taken from the various links and buttons that appear on the Clients page. And you can use the Accounts page as your starting point to explore more in-depth client information, view client projects, view client estimates, and more. Now, we'll take a closer look at the setup of the Clients page, and the range of actions available to you on the Accounts page.

To view your client list page, go to the main sidebar and click the Accounts tab.

11.2.1 Overview

The Clients page presents a list summary of all your current clients in a table format. The main elements of the table include:

- **Name:** The name of the client

- **Contact Person:** The name of the primary contact person
- **Email:** The client email address
- **Balance:** The client's payment balance
- **Expenses:** The client's unbilled expenses

11.3 Create Client

So, you've taken on a new client? Congratulations!

Your Clients list is at the heart of your invoicing activity, so it's really important to maintain current information on all your clients. When you start working with a new client, the first thing you'll need to do is to add the new client by entering their contact information and business details.

When creating and saving a new client to your Client list, make sure to have the relevant, up-to-date information at hand. You are only required to enter the information one time. WorkiceCRM automatically tracks all invoicing,projects,estimates and payment activity for each client.

Client Creation

To create clients individually.

In the Clients module, click (Create button) to add a new account.
In the Create Client page, enter the account details.
Click Save

Tip: The Create Client page is divided into four sections. Enter the information in the relevant fields.

Let's take a closer look at each section:

- **General**: Enter details about your client's business/company/organization, including the company name, email, contact person, tax number and tags (optional).
- **Contact**: Enter contact details related to your client including phone number, address, currency, preferred locale and a logo (optional).
- **Web**: Enter your client social media or website data.
- **Custom Fields**: If there are any custom fields for clients, it will be displayed here.

11.4 Client Overview Page

Once you have created an account, you can view the details of the account in the record's details page. The details page of an account presents information related to the account - for example, contacts, projects, invoices, deals, payments, expenses, subscriptions and so on - in a single location.

Tip: If you entered the client's street address, a Google map appears below the information box displaying the client's location.

11.4.1 Invoices Section

The Invoices section shows a list of all the client's invoices and accompanying information.

- **Invoice Reference**: The invoice number
- **Date Issued**: The date the invoice was created
- **Amount**: The invoice amount
- **Balance**: The invoice balance
- **Due Date**: The date the payment is due
- **Status**: The status of the invoice (Draft, Not Paid, Sent, Viewed, Paid, Overdue)

Tip: You can also create a new invoice for this client via the Create button that appears at the top left of the Invoices section.

11.4.2 Payments Section

The Payments section shows a list of all the client's payments and accompanying information.

- **Transaction reference**: The reference number of the transaction
- **Method**: The payment method (ie. Paypal, Stripe, Cash, etc)
- **Amount**: The payment amount
- **Date**: The date the payment was made
- **Currency**: Currency used in the payment

11.4.3 Projects Section

The projects section shows a list of all the client's projects and accompanying information.

- **Title**: The invoice number
- **Expenses**: Total expenses for the project
- **Start Date**: The start date of the project
- **Amount**: Total cost of the project
- **Progress**: Project progress in percentage
- **Due Date**: The date the project is due
- **Status**: The status of the project (Active, Done, On Hold)

11.4.4 Estimates Section

The estimates section shows a list of all the client's estimates and accompanying information.

- **Estimate Reference**: The estimate number
- **Date Issued**: The date the estimate was created
- **Amount**: The estimate amount

- **Viewed**: An icon showing whether an estimate has been viewed by client
- **Due Date**: The date the estimate is due
- **Status**: The status of the estimate (Draft, Pending, Sent, Approved, Rejected, Overdue)

Tip: You can also create a new estimate for this client via the Create button that appears at the top left of the estimates section.

11.4.5 Expenses Section

The expenses section shows a list of all the client's expenses and accompanying information.

- **Expense reference**: The reference number of the expense
- **Category**: Expense category (ie Housing, Marketing etc)
- **Amount**: The expense amount
- **Date**: The date the expense was incurred
- **Invoiced**: Whether the expense has been invoiced
- **Currency**: Currency used in the expense

11.4.6 Files Section

The Files section shows a list of all the client's associated files.

11.4.7 Subscriptions Section

The subscriptions section shows a list of all the client's subscriptions.

11.4.8 Updating Client

Click on the Update button, at the top right corner of the page. You will now be taken to the Account/Update page, where you can edit any of the fields.

11.4.9 Deleting the Client

You can also delete the specific client directly from their Client Overview page. Click on the trash icon at the right hand side of the Update Client button.

CHAPTER 12

Invoices

Invoices allow you to bill a Client for your products and/or services, and help you keep track of your income in Workice CRM. Every invoice paid means more revenue coming into your business. Create and send professional invoices to your customers in seconds. Once you've entered the client and tax information, you'll have a range of actions at your fingertips – from saving a draft, to sending the invoice to the client via email, to printing a PDF hard copy.

12.1 Overview

The life of an invoice in Workice system is made up of a number of stages:

- **Draft/Hidden**: When you've created an invoice, but have not yet sent it.
- **Sent**: You've sent the invoice, but the client has not yet paid.
- **Viewed**: The client has opened the invoice email and viewed the invoice.
- **Partial**: The invoice has been partially paid.
- **Paid**: Congratulations! The client has paid the full invoice amount.
- **Not Paid**: The invoice remains unpaid.
- **Overdue**: The invoice has passed its due date.

12.2 Invoices list

On the invoices list page you'll see a table with the columns below;

- **Invoice #**: The number of the invoice
- **Client Name**: The name of the client
- **Status**: The current status of the invoice (Draft, Sent, Viewed, Partial, Paid, Not Paid, Overdue)
- **Due Date**: The date the payment is due

- **Amount**: The total amount of the invoice
- **Balance**: The amount owed by the client (after credits and other adjustments are calculated)

12.3 Create Invoice

To create a new invoice, go to the Invoices tab on the main sidebar, and click on the + Create button. This will open the Invoices / Create page offering a series of text and numerical inputs.

Note that each new invoice you create will be automatically numbered in chronological order. This will ensure your records are kept logical and organized. (You have the option to change the invoice number manually in **Settings - Invoice Settings**).

The form contains:

- **Ref No**: Auto assigned invoice number
- **Title**: Invoice Title e.g Acme Website Design (optional)
- **Client**: Click on the arrow at the right end of the Client field. Select the relevant client from the client list.
- **Tax 1**: Tax 1 amount in percentage.
- **Tax 2**: Tax 2 amount in percentage.
- **Discount**: Discount as percentage or monetary amount.
- **Late Fee**: The percentage or amount to be applied if the invoice is not paid on time.
- **Extra Fee**: If there is any additional fee for the invoice add it as amount or percentage.
- **Currency**: Invoice currency. If you select `Client Default Currency`, the selected client's currency will be used.
- **Tags**: Enter multiple tags for the invoices e.g website,logos etc (Optional)
- **Payment methods**: Select the payment methods you want your client to pay with. All enabled payment methods will be presented to the client as options so he/she can choose the payment method to use.

Tip: You can create a new client while creating a new invoice. Simply click on the Create new client link, situated on the top left side of the Create page. A pop-up modal will open, enabling you to complete the new client's details. Then continue creating the invoice for this new client.

Attention: If you use Braintree, you'll need to enter your Merchant Account.

- **Partial Payment Terms**: If you want to enabled partial payments, enter the phases that the invoice should be paid with their due dates. Example, 50% - Due 01-01-2019 and 50% - Due 02-02-2019. If you require an invoice to be paid once, enter 100% in the amount and the deadline/due date.
- **Notes**: Want to enter information to appear as a footer on the invoice? Enter it here. The text will appear at the bottom of the invoice.

Once you've completed creating your invoice, click on **Save** Button and you'll be redirected to the invoice page where you can enter your products/services to bill your clients.

- **Item/Product**: This is the name of the item you are billing for. You can either enter the details manually, or start typing and pick already invoiced items.

- **Description**: Add more information about the item. This will help the customer better understand the job completed, and is also useful for your own reference.
- **Unit Price**: The amount you charge per unit of items. For example, let's say your item is "1 hour consulting", and you charge $80 for an hour of consulting – that is, for 1 item unit. Then you'll enter 80 in the Unit Price field.

Note: If you have selected a set item from the auto complete list, the description and unit price that you pre-defined in your previous invoice will apply by default. You can manually override the default unit price or description by clicking in field and changing the data.

- **Quantity**: The number of units being charged. Continuing the above example, let's say you need to charge for 3 hours of consulting, enter the number 3 in the Quantity field.
- **Tax Rate**: Note: To apply tax to the line item, click on the arrow at the right side of the Tax field and select the relevant tax from the drop-down list.
- **Discount**: This is the discount percentage you need to apply for the particular line item.

Click on **Save** button to save the item.

Beneath and to the right of the line item section, you'll find the Totals section:

- **Subtotal**: This is the amount due before other figures are taken into calculation, such as Tax, Discounts, Credits, etc.
- **Tax 1**: Tax 1 rate for the invoice.
- **Tax 2**: Tax 2 rate for the invoice.
- **Payment Made**: The amount paid to date, including partial payments and credits.
- **Balance**: The final balance owed to you, after taxes, partial payments and credits have been deducted from the charged amount.

12.4 Invoice Page

- **Show to client button**: Use this button to hide/show invoice to client.
- **Pay Invoice button**: Click this button to make payment to an Invoice.
- **Email button**: Email the invoice directly via Workice system to the email address specified for the client.
- **Activity button**: Click to view invoice history.
- **Set Reminder button**: Add custom reminder and get alert. e.g Reminder to send invoice
- **Comments button**: Add invoice comments here.
- **More button**: Access additional invoice options including updating, deleting invoice.
- **Mark Sent**: When you mark an invoice as sent, only then is the invoice viewable to the client in the client portal, and the client balance is updated to reflect the invoice amount.
- **Mark Paid**: Manually mark the invoice as paid. You may want to do this if you are not entering the payment directly into the system.
- **Delete Invoice**: Click here to delete the invoice. It will be deleted and removed from the Invoices list page.
- **Share button**: Dispays a link that you can send to client to access the invoice.
- **PDF button**: Download a PDF version of the invoice.

- **As Client button**: You can impersonate a client and view the invoice as client.

12.5 Email Invoice Preview

When you are ready to send an invoice to the client, click the Email Invoice button. Before the invoice email is sent, a pop-up box will open, displaying a preview of the email. Here, you can add additonal comment to the email.

12.5.1 Customizing the Invoice Email Template

To customize the email template, go to **Settings** - **Translations** and click on **Emails button** on the top navigation and select the locale you want to modify.

Tip: You can customize any type of email template, including invoice emails, First Reminder, Second Reminder and Third Reminder emails. The english version variables are named in **module**, **action** and **message** format (dot notation). Example; if you need to edit the message that will be sent when you send an invoice, look for a variable named `invoices.sending.body`. To edit sent message subject, modify `invoices.sending.subject` value.

12.6 Instant Notification

Know when an invoice is viewed, becomes due, or gets paid, so you can take the right actions to manage your cash flow. Set up invoice reminders to automatically email your customers when payment is due.

12.7 Reuse items as much as you want

Recycling is a good thing, so why waste time and effort writing in the same items and prices over and over again? Once you add your items to an invoice you'll only need to start typing in your invoices to see them pop up.

12.8 Auto Reminders

Save yourself the time and hassle and automate your client communications! An invoice reminder is an automatic email message to remind your customer that an invoice is coming due or that it is overdue. This is a great way to stay on top of reminding your customers that you should be getting paid soon.

You can have Workice CRM send reminders that invoices will be due to be paid soon, and we call these **Upcoming Reminders**. You can also have Workice send reminders that invoices are overdue, and we call these **Overdue Reminders**.

12.8.1 Upcoming Reminders

To send invoice reminders before the due date, modify **Settings** > **Invoice Settings** > **Upcoming reminders**. Default is 3 days

- Change to the number of days you want a reminder to be sent before the due date.
- Example; Setting it to 2 days will send invoice reminders 2 days before invoice overdue date.

12.8.2 Overdue Reminders

To enable Invoice Overdue Reminders, go to **Settings** > **Invoice Settings** > **Send Overdue Emails** and enable it.

Once you've got these reminders set up, you don't have to do anything else; Workice will continue to send these reminders on the schedule you set until the end of time (or until you get paid, whichever comes first).

Tip: Modify the number of days to send each invoice reminder in **Settings** - **Invoice Settings** section. You may also set late fee to apply on third reminder.

12.9 Recurring Invoice

As a busy freelancer, you work for a variety of clients. Some jobs are one-off, but others are ongoing, whether on a weekly, monthly or other basis. Workice CRM recurring invoice feature automatically creates invoices for ongoing jobs, and sends the current invoice to the client on a regular, pre-defined basis. For each recurring job, you only need to set up the procedure once.

To make a invoice recur, edit the invoice and select the **Recur Every** dropdown. You can set it to recur every `week, month, quarter, six months and yearly`. Select the start date and a date when the invoice should stop recurring (End Date).

Tip: To stop a recurring invoice, edit the invoice and change **Recur Every** field to **None**.

Tip: Reminders are sent based on the due date of the invoice.

Tip: To disable/enable sending invoices immediately they recur, go to **Settings** - **Invoice Settings** and disable/enable **Email on Recur** checkbox..

When the invoices from this invoice will be generated you will have an overview which invoices are generated from this invoice at the Child Invoices link on the invoice page.

12.10 Apply Credit Notes

When an invoice has recurred, you can choose to apply client credits to the invoice by enabling this feature in **Settings** > **Invoice Settings** > **Apply Credits**.

12.11 Bulk Actions

If you need to perform an action for a number of invoices, you can do it in one click with the bulk action feature. To use the bulk action feature, mark the relevant invoices in their checkbox at the far left of the invoices list. Once you've marked the invoices, select an action to perform on them in the buttons below the invoice list page.

- **Send**: Send selected invoices by email to client(s).
- **Mark as Paid**: Mark selected invoices as paid.

- **Archive**: Archive selected invoices.
- **Delete**: Delete selected invoices.

CHAPTER 13

Payments

Workice CRM handles your entire freelance invoicing process – from sending estimate, to invoicing your client, to receiving payment. What's more, you can receive payments directly and automatically via workice supported gateways enabling totally smooth management of your customer accounts using your choice of payment provider.

13.1 Payments List

The Payments list page displays a summary of all payments once they have been received. Payments are recorded in two ways:

1. **Automatic payment**: If your client has paid you via any of the supported payment gateways, the payment will be automatically recorded in the Payments list. You will be notified on your dashboard page in the notification section, and also via email (if you have enabled notification).
2. **Manual payment**: If your client has paid you via cash, check, bank transfer, credit card or any other payment system not linked to Workice CRM, you will need to enter the payment manually on the Payments / Create page.

Whether automatic or manual entry, the Payments list page presents an overview of all payments received in a user-friendly table format. Now, we'll take you through the various columns in the Payments table from left to right:

- **Code**: Transaction reference number.
- **Client Name**: The client's name
- **Payment Date**: The date the payment was received
- **Invoice Date**: The date the invoice was issued.
- **Amount**: The payment amount that was received
- **Method**: The method of payment used, ie. PayPal, Bank Transfer, Visa, etc

13.2 Partial Payments

Accepting partial online payments is simpler than you think. Your customer has the option to enter an amount less than the invoice amount while making online payment. Once the amount is successfully processed, it's applied to the invoice and the status of the invoice is updated to "Partially Paid". There 3 options a client can pay a partial payment;

1. **Minimum Due**: The customer can choose to pay minimum amount due to keep the account current.
2. **Full Amount**: Client can pay the full invoice amount
3. **Other Amount**: Customer enters the amount to pay in the input.

13.3 Refunding a Payment

If you need to refund a payment, go to the relevant payment in the Payments list, click on the payment. Click on **Refund** button and the transaction will be marked as refunded.

13.4 Payment Receipt

You can download a payment receipt in the payment overview page by clicking on **Receipt** button.

13.5 Manually Creating a New Payment

To create a new payment, go to Invoices list page and select the invoice to pay. Click on **Pay** button and a number of fields will be displayed;

- **Amount**: The invoice amount will appear automatically by default. However, if the payment amount does not correspond to the default invoice amount, you can manually Enter the amount of payment received.
- **Payment Date**: The date the payment was received.
- **Payment Type**: Select the payment method that was used. Select the appropriate method from the list. Options include Bank Transfer, Cash, Razorpay, Braintree, Stripe, PayPal, check and more.
- **Receipt**: Check this option to enabled upload of the Payment Receipt.
- **Send Email**: Enable it to send a thank you email.
- **Notes**: Here, you can add any comments or notes.

13.6 Bulk Actions

If you need to perform an action for a number of payments, you can do it in one click with the bulk action feature. To use the bulk action feature, mark the relevant transactions in their checkbox at the far left of the payment list. Once you've marked the transactions, select an action to perform on them in the buttons below the payments list page.

- **Archive**: Archive selected transactions.
- **Delete**: Delete selected transactions.

CHAPTER 14

Creditnotes

14.1 Credits List

In some cases, the customer no longer wants the item or service ordered and you are needing to apply a credit or refund the customer, this is where the Credits feature comes in handy.

14.1.1 Overview

The Credits list page is a summary of all credits issued to all clients and contains the columns below;

- **Client Name**: The client's name
- **Status**: The credit note status (e.g Open, Closed, Void)
- **Date**: The date the individual credit was issued
- **Amount**: The credit note total amount
- **Balance**: The balance of the individual credit

14.2 Enter Credit

To issue a credit, you'll need to open the creditnotes create page by clicking **Create** button.

The create credits page has four fields for you to complete. Let's take a closer look at each field:

- **Client**: Click on the arrow at the right end of the Client field. Select the relevant client from the client list.
- **Currency**: The currency used for this creditnote
- **Tags**: Custom tags (optional)
- **Terms**: [Optional] Enter any terms (Will appear at the bottom of the credit)

To save your new credit, click **Save**.

14.3 Apply Credits

To apply a credit to the customers invoice for the amount they paid: Access the customer invoice that needs to be credited and select More in the top bar of the Invoice page. Select **Use Credit** in the drop down. Here, you can enter the amount to apply in the textbox. Click on **Save** button to use credits.

Now that the credit has been applied to the invoice, you will see the credit reflected in the customers account, as well as the invoice.

14.4 Bulk Actions

If you need to perform an action for a number of creditnotes, you can do it in one click with the bulk action feature. To use the bulk action feature, mark the relevant credits in their checkbox at the far left of the credits list. Once you've marked the creditnotes, select an action to perform on them in the buttons below the credits list page.

- **Send**: Send selected credits by email to client(s).
- **Archive**: Archive selected creditnotes.
- **Delete**: Delete selected creditnotes.

Tip: Need to search for a specific client in your Credits list? Start typing the first letters of the client's name and the filter will automatically present the relevant listings.

CHAPTER 15

Projects

To help manage costs you can set project budgets, record expenses and track timesheets against the project and tasks. If you're charging a customer for your work, you have the flexibility to charge fixed amounts as work is completed and/or set time-based billable rates. Hourly rates can be set for each team member.

15.1 Projects list

On the projects list page you'll see a table with the columns below;

- **Name**: Project title/name
- **Client**: The name of the client
- **Team**: Team members attached to the project
- **Budget**: Percentage used budget (Uses project estimated hours)
- **Total Time**: Total time spent on the project
- **Amount**: The current total cost of the project
- **Expenses**: Total expenses for the project

15.2 Create Project

To create a new project, go to the Projects tab on the main sidebar, and click on the **+Create** button. This will open the project create page offering a series of text and numerical inputs.

The form contains:

- **Name**: Project title/name
- **Client**: Select the relevant client from the client list.
- **Progress**: Client progress (0-100)

- **Team**: Select team members.
- **Start Date**: Project start date
- **Due Date**: Expected end date for the project
- **Currency**: Currency used in the project
- **Billing Method**: There are four billing methods and each one is explained below;
- **Est. Hours**: Estimated number of hours for the project.
- **Description**: Summary of the project.
- **Project Settings**: Per project settings to disable/enable on client section.
- **Tags**: You can assign multiple tags to projects by entering them directly in the provided field e.g design,logos etc.

Tip: You can create a new client while creating a new project. Simply click on the **New Client** link, situated on the right side of the Client field. A pop-up modal will open, enabling you to complete the new client's details. Then continue creating the project for this new client.

15.2.1 Billing Methods

Hourly Staff Rate

If you have Staff in your account, their rate is how much your Client will be billed regardless of the Task the hours are logged against. This method is really useful if you have multiple Staff working on a Project all with different rates.

Hourly Task Rate

Your Client will be billed based on the Tasks that are assigned to the Project. It doesn't matter who tracked the time, only the **Task Rate** is being taken into consideration. This method is really useful if you charge different rates for different Tasks regardless of who did the work.

Hourly Project Rate

It doesn't matter who tracked the time, or which Task was chosen, your Client will always be billed at one consistent rate for that Project.

Fixed Project Amount

It doesn't matter how many hours were logged, the Project will always be billed at the Flat Project Amount. You set the Flat Project Amount from the Edit Project screen.

15.3 Time Entries

15.3.1 Manual Entry

- To enter time manually click on **Time Entry** button.

- Select a task from the dropdown list (optional).
- Enter time in the format {hours}:{minutes}:{seconds} i.e 01:10:30 means 1 hour, 10minutes and 30 seconds.
- Alternatively, you can choose a start time and end time by checking **Use start and End Dates** checkbox.
- Enter time entry description and choose whether it's billable or unbillable.

Click **Save** when completed.

15.3.2 Start Timer

To start the project timer.

- Click on the **green stop watch icon** at the top bar of the project overview page.
- Running timers will be displayed on the top navigation section of the CRM.
- You can stop the timer by clicking on the **red spinning clock icon** that appears at the top bar of the project overview page.

15.4 Invoice Project

In Workice CRM, you and your team can track your time to Projects. Those Projects are assigned to individual Clients, and you can Invoice your Clients easily based on that tracked time by generating an Invoice with time entries on it.

To generate an invoice.

- Open the project dashboard and click on the **ellipsis/more button (. . .)** at the top bar of the project page.
- Click **Invoice** to start invoicing your project.
- The total unbilled time will be displayed on the modal.
- If there are unbilled expenses, they will be listed for inclusion.
- Choose how you want the time entries to be formatted (displayed) on your Invoice – you can select Task per line or Single Line.

Click **Save** to generate project invoice.

Tip: Billed time entries will all be marked as **Billed**.

15.5 Duplicate Project

To duplicate a project, click on the **ellipsis/more button** and select elements to duplicate i.e tasks, expenses,comments, issues etc. A new project will be generated similar to the duplicated project.

15.6 PDF Preview

You can download project summary as PDF by clicking on **ellipsis/more button** and select **PDF**. A PDF document will be downloaded with the projects summary i.e Time Entries and Tasks.

Tip: To mark a project as complete. Click **Done** accessible via the **ellipsis/more button**.

15.7 Bulk Actions

If you need to perform an action for a number of projects, you can do it in one click with the bulk action feature. To use the bulk action feature, mark the relevant projects in their checkbox at the far left of the parojects list. Once you've marked the projects, select an action to perform on them in the buttons below the projects list page.

- **Invoice**: Invoice selected projects.
- **Archive**: Archive selected projects.
- **Delete**: Delete selected projects.

CHAPTER 16

Subscriptions

Subscriptions feature helps you bill customers for your services once or on a recurring basis. Effortlessly manage the entire customer life cycle, from onboarding a customer for a subscription plan to accepting payments. Once a subscription has been created, it will begin to auto-renew on the period specified by the product to which it is related. There's not much you need to do, other than keep tabs on them from time to time looking for subscriptions that go sour.

Attention: Subscription payments is provided via Stripe you will need to have active Stripe account in order to use this feature. More subscription integrations will be included in future releases.

Attention: For subscriptions to work, you need to configure Stripe Keys. Check **Configure** under installation section of this documentation for more details.

Tip: If you haven't created your products and billing plans, you should create via the Stripe Dashboard, Workice CRM will fetch the billing plans directly from Stripe and will display them while creating/updating subscription.

16.1 Getting Started

Start by creating your stripe products on Stripe Dashboard.

- Click on **Create** button in subscriptions list page.
- If you have setup your Stripe configuration correctly, you should get a popup modal with the fields below;
- **Name** - Subscription name that will be shown to the customer
- **Billing Date** - First billing date field, leave blank to use the date when the customer is subscribed to the subscription.

- **Client** - The name of the client
- **Stripe Plan** - Retrieved from your stripe account.(Your Stripe Products)
- **Description** - Subsscription description

Attention: To view subscriptions that a client has not subscribed, click on **Plans** button.

After you have created a subscription for the client, click on **Plans** table to view the subscription, modify, delete or send it to your customer. Your client will see the subscription on their dashboard where they can subscribe.

Tip: A new role named `subscriber` will be created/attached to the customer primary contact.

16.2 Cancel Subscription

If you want to cancel subscription from the admin section, you can open the subscription and click on the cancel button;

16.2.1 Cancel Immediately

If you click Cancel Immediately, the subscription will be cancelled immediately and not option to reactivate.

16.2.2 Cancel on expiry

If the Cancel Immediately checkbox is left unchecked, the subscription will cancel at the end of billing period and you’ll have an option to re-activate it.

16.3 Subscription Invoices

Your client/customer can download their invoices by clicking on **Invoices** button at the top bar of subscriptions list page.

CHAPTER 17

Knowledgebase

Internet businesses are, in a sense, always open. Customers interact with your website 24/7, and when they have a question, they want an answer that minute.

The best knowledge software anticipates your users' most common problems and makes it quick and easy for them to find the solution they need. That's why a knowledge base is also useful for customer service agents. While working on a user's ticket, an agent can interact with your intuitive knowledge base content to learn more about their problem.

17.1 Create Categories

To create article categories, click on **gears** icon at the top right of the articles list page. A modal will pop up and you can create, edit or delete article categories.

17.2 Create Article

- Click on **Create** button
- Add article **subject** and select a **category**
- Write the body of your article.
- To disable comments on the article, uncheck **Allow Comments** checkbox
- To save the article as draft/unpublished, uncheck the **Published** checkbox

> **Attention:** The editor supports markdown text do not use html tags.

CHAPTER 18

Estimates

Estimates allow you to create and send an approximate outline of your Items or Freelancer Services to your Client for approval before beginning the work. The client can then Accept the Estimate for approval before converting it into an Invoice automatically, or you can go ahead and convert it yourself to start collecting payment!

18.1 Estimates List

As a freelancer, many jobs – whether big or small – will begin as an estimate. How so? Well, a client expresses interest in your skills or product, but wants to know how much it's going to cost, in advance. With the estimates feature of Workice, you can easily create a price estimate and send it to the client up front, helping them make the decision to commit, and helping you keep track of your projected work schedule and income.

Once the price quote is accepted by the client, Workice enables automatic creation of an invoice that imports all the estimate data, so your estimates and invoices are linked, organized and easily traceable.

To view your estimates list page, click the **Accounting** tab on the main sidebar, and select Estimates from the drop-down menu. This will open the Estimates list page.

18.1.1 Overview

The estimates list table contains the columns below:

- **Estimate**: The number of the estimate
- **Client**: The client name
- **Status**: The current status of the estimate (Pending, Approved, Invoiced)
- **Due Date**: The last date that the estimate is valid and can be accepted by the client
- **Amount**: The total amount of the estimate.
- **Created At**: The date the estimate was created

18.2 Create Estimate

To create a new estimate, go to the Estimates tab on the main sidebar, and click on the **Create** button. This will open the Estimates / Create page offering a series of text and numerical inputs.

Note that each new estimate you create will be automatically numbered in chronological order. This will ensure your records are kept logical and organized. (You have the option to change the estimate number manually in **Settings** - **Estimate Settings**).

The form contains:

- **Title**: Estimate Title e.g Acme Website Design (optional)
- **Client**: Click on the arrow at the right end of the Client field. Select the relevant client from the client list.
- **Due Date**: The date when the estimate should expire.
- **Currency**: Estimate currency.
- **Discount**: Applying a discount to an estimate is the same as applying a discount to an invoice.
- **Tax 1**: Tax 1 amount in percentage.
- **Tax 2**: Tax 2 amount in percentage.
- **Tags**: Enter multiple tags for the estimates e.g website,logos etc (Optional)
- **Notes**: Want to enter information to appear as a footer on the estimate? Enter it here. The text will appear at the bottom of the estimate.

Tip: You can attach a new deal or existing deal to an estimate. Once an estimate is approved, the deal is marked as Won.

Once you've completed creating your estimate, click on **Save** Button and you'll be redirected to the estimate page where you can enter your products/services.

- **Item/Product**: This is the name of the item you are billing for. You can either enter the details manually, or start typing and pick already invoiced items.
- **Description**: Add more information about the item. This will help the customer better understand the job completed, and is also useful for your own reference.
- **Unit Price**: The amount you charge per unit of items. For example, let's say your item is "1 hour consulting", and you charge $80 for an hour of consulting – that is, for 1 item unit. Then you'll enter 80 in the Unit Price field.

Note: If you have selected a set item from the auto complete list, the description and unit price that you pre-defined in your previous invoice will apply by default. You can manually override the default unit price or description by clicking in field and changing the data.

- **Quantity**: The number of units being charged. Continuing the above example, let's say you need to charge for 3 hours of consulting, enter the number 3 in the Quantity field.
- **Tax Rate**: Note: To apply tax to the line item, click on the arrow at the right side of the Tax field and select the relevant tax from the drop-down list.
- **Discount**: This is the discount percentage you need to apply for the particular line item.

Click on **Save** button to save the item.

Beneath and to the right of the line item section, you'll find the Totals section:

- **Subtotal**: This is the amount due before other figures are taken into calculation, such as Tax, Discounts etc.
- **Tax 1**: Tax 1 rate for the estimate.
- **Tax 2**: Tax 2 rate for the estimate.
- **Discount**: The discount amount.
- **Total**: The total amount for the estimate

18.3 Estimate Page

- **Show to client button**: Use this button to hide/show estimate to client.
- **Activity button**: Click to view estimate history.
- **Set Reminder button**: Add custom reminder and get alert. e.g Reminder to send estimate
- **Comments button**: Add estimate comments here.
- **More button**: Access additional estimate options including updating, deleting estimate.
- **Send button**: Email the estimate directly via Workice system to the email address specified for the client.
- **Mark as Accepted**: The estimate status will change to accepted.
- **Mark as Declined**: The estimate status will change to declined.
- **Delete**: Click here to delete the estimate. It will be deleted and removed from the estimates list page.
- **Share button**: Dispays a link that you can send to client to access the estimate.
- **PDF button**: Download a PDF version of the estimate.

18.3.1 Convert to Invoice

Once an estimate has been accepted, you can click on **Convert** button to convert it to an Invoice.

Note: You can set an estimate to convert to an invoice autmatically in **Settings** -> **Estimate Settings**.

18.3.2 Convert Estimate to Project

To convert estimate to project once approved by the client. Go to **Settings** -> **Estimate Settings** and check **Estimate to Project** checkbox.

Tip: You may attach estimate documents using the folder icon at the top right side of the invoice top navigation..

18.4 Email Estimate Preview

When you are ready to send an estimate to the client, click the **Send** button. Before the estimate email is sent, a pop-up box will open, displaying a preview of the email. Here, you can add additonal comment to the email.

18.4.1 Customizing the Estimate Email Template

To customize the email template, go to **Settings** - **Translations** and click on **Emails button** on the top navigation and select the locale you want to modify.

Tip: You can customize any type of email template, including invoice emails, First Reminder, Second Reminder and Third Reminder emails. The english version variables are named in **module**, **action** and **message** format (dot notation). Example; if you need to edit the message that will be sent when you send an estimate, look for a variable named `estimates.sending.body`. To edit sent message subject, modify `estimates.sending.subject` value.

18.5 Instant Notification

Know when an estimate is viewed or becomes due so you can take the right actions to manage your cash flow.

18.6 Reuse items as much as you want

Recycling is a good thing, so why waste time and effort writing in the same items and prices over and over again? Once you add your items to an invoice/estimate you'll only need to start typing in your estimates to see them pop up.

18.7 Bulk Actions

If you need to perform an action for a number of estimates, you can do it in one click with the bulk action feature. To use the bulk action feature, mark the relevant estimates in their checkbox at the far left of the estimates list. Once you've marked the estimates, select an action to perform on them in the buttons below the estimates list page.

- **Send**: Send selected estimates by email to client(s).
- **Archive**: Archive selected estimates.
- **Delete**: Delete selected estimates.

CHAPTER 19

Contracts

One of the ways of building a relationship between a freelance developer and a client is to offer e-contracts built into your web interface. e-contracts is a movement towards a faster and more seamless contract process and experience.

- The default contract clauses included in Workice was written by AND.CO and is vetted by the Freelancers Union and automatically complies with the Freelance Isn't Free Act.
- You and your clients can sign contracts online.
- Get alerts when the contract is viewed or signed.

19.1 Create Contract Template

Start by creating a contract template, Go to Contracts and click on Templates button.

- Enter the name of the contract template e.g Web Design Contract
- Enter the contract clauses (check sample contract included on installation)
- A list of available tags is shown below the Contract Message textarea. Those tags will be replaced with their values example {EXPIRY_DATE} will be replaced with the date when the contract expires.
- After creating a contract template, you can select the Contract Template when creating a contract as shown in the steps below;

19.2 Create Contract

To create a contract, Go to **Contracts** and click on **Create** button.

You will be redirected to contract creation form where you can enter contract information.

- **Contract Title**: - The Contract Title
- **Client**: - Client associated with the contract

- **Contract Template**: - The contract template to be used e.g Web Design
- **Start Date**: - Contract start date
- **End Date**: - Contract end date
- **Fixed Rate**: - Select **Yes** to set a fixed rate for the contract
- **Fixed Price**: - If the above field is set to **Yes** enter the fixed price in monetary value.
- **Hourly Rate**: - Set the contract hourly rate
- **Currency**: - The currency to use in the contract
- **Services**: - A comma separated list of services e.g web design, consulting etc
- **Who will own your work**: - Select who owns the copyright of your work.
- **How long should client have to countersign the contract**: - number of days the client has to sign the contract. e.g 14 for 14 days
- **Description**: - Contract description text
- **Payment Terms**: - Number of days a client has to pay for each invoice
- **Late Payment Fee**: - Late payment fee as percentage/monetary value
- **Late fee is percent**: - Whether the field above (Late Fee) is a percentage/monetary value
- **Project Termination Notice**: - Number of days a client should notify you on contract termination
- **Cancellation Fee**: - Amount to charge a client incase they terminate the contract without cause
- **Required Deposit**: - Amount of money to be paid upfront
- **Right to include in Portfolio**: - Right to include work attribution in your portfolio
- **Add Non-Compete**: - Add Non-Compete clause
- **Revision Requests**: - Number of allowed revision requests
- **Sexual harassment clause**: - Add sexual harassment clause

Once you are done entering the contract information, click **Save** to generate a contract for your client.

19.3 Contract Page

19.3.1 Sign and Send

To sign the contract, click the green **Sign and Send** button and type your name. Once you have signed the contract, Workice will send an email to client to review the contract and sign.

19.3.2 Contract PDF

Click on **PDF** button to doanload PDF version of the contract.

Tip: Click **Share** button to get a link that you can send to your staff/customer.

Note: You will receive an email once your client views the contract.

CHAPTER 20

Tasks

Projects are made up of tasks, and knowing how to manage your tasks (and everyone else's) is the secret to getting your projects completed on time. The ability to keep track of your tasks and deadlines makes the task management feature a crucial tool for your freelance business needs.

20.1 Tasks List

The Tasks list displays a list of all your tasks, for all projects, in an easy-to-follow table format.

Tip: You can mark a task as completed by simply clicking on the checkbox that appears on the left side of the task name.

20.2 Create Task

20.2.1 Stages

To create project task stages, open a project and click on tasks tab. Click on the **gears** icon on the right side of the top navigation bar. A modal will popup that allows you to add, update or delete the stages.

20.2.2 New Task

To create a new task, open the project you want to create a task for. Click on **tasks** tab and select **Create** button.

The task creation form contains the fields below;

- **Task Name**: Enter the task name
- **Milestone**: Select a milestone attached to this project (optional)

- **Stage**: Enter the task stage e.g Backlog
- **Team**: Select multiple users assigned to the task
- **Description**: Enter a task description
- **Start Date**: Date when the task should start
- **Due Date**: Date when the task should end
- **Hourly Rate**: The blling rate to charge per hour for the task
- **Est. Hours**: Total number of estimated hours a task takes
- **Recur Every**: Select the frequency in which the task repeats.
- **Visible to Client**: Uncheck it to hide from client
- **Tags**: Custom tags e.g logos, design etc

20.3 Time Entry

There are two ways to record time spent on a task: Timer and Manual.

20.3.1 Timer

This is an automatic timer feature that allows you to record working sessions in real time. To begin the timer, press Start. Then get to work. The timer will run until you select Stop. When you select Stop, the timer will be stopped and the time entry can be viewed in **Timesheets** section of the task page.

20.3.2 Manual

The other way you can create a time entry is by manual entry of the session details.

- To enter time manually click on **Time Entry** button.
- Select a task from the dropdown list (optional).
- Enter time in the format {hours}:{minutes}:{seconds} i.e 01:10:30 means 1 hour, 10minutes and 30 seconds.
- Alternatively, you can choose a start time and end time by checking **Use start and End Dates** checkbox.
- Enter time entry description and choose whether it's billable or unbillable.

20.4 Recurring Tasks

You can create recurring tasks for tasks you have to do again and again as part of a single project. Before setting a repeat, you will first need to make sure you have a due date set on the task.

When editing the task, you will see the **Recur Every** field and under this field you can choose when the task should repeat:

You can set a repeat interval such as `daily, week, month, quarter, six months, year`.

Once you have chosen the interval, you can decide for the repeat to stop by a particular date.

Note: Any files attached to the original task will not carry over to each future repeating task.

Tip: To stop a recurring task, edit the task and change **Recur Every** field to **None**.

Tip: Reminders are sent based on the due date of the task.

CHAPTER 21

Expenses

Track all of your lunch meetings or training materials, and other miscellaneous expenses in the Expenses section, and instantly bill clients. Expenses allow you to capture the billable and non-billable costs that you incur as part of your project work. These non-labor expenses are usually itemized costs for things like materials, travel expenses, or fixed-fee services. Typically, billable expenses are passed on to a client or customer, whereas non-billable expenses are internal costs paid for by your employer.

Tip: You can create Expenses that you incur as part of your business operations (for example, Internet Expenses), or Recurring Expenses for those charges you incur on a frequent basis.

21.1 List Expenses

To view the Expenses list page, click on the Expenses tab on the main sidebar.

21.1.1 Overview

The Expenses list page displays a list of all business expenses that you choose to enter. The table columns appear as below;

- **Code**: Reference number for the expense
- **Client**: The name of the client for whom the expense is relevant
- **Project**: The name of the project attached to the expense
- **Amount**: The expense amount including taxes
- **Billed**: Whether the expense has been billed to client
- **Category**: The assigned category of the expense
- **Expense Date**: The date the expense occurred

21.2 Expense Categories

Workice makes it easy to keep track of your spending with Expenses with categories to organize them.

Workice also makes it easy for you to create your own custom categories if a specific one isn't available.

21.2.1 Adding/Editing Categories

Click on the **Gears** icon at the topbar of the expenses list page. A modal will popup with a list of all existing categories. Here you can add, edit and delete your expenses categories.

21.3 How Taxes Work on Expenses

> **Attention:** When creating an Expense with a tax on it, the system will automatically calculate taxes for you based on the total amount you paid.

Here's a sample tax situation:

You paid $113.00 in gas (including tax) while driving around the city.

- Enter $113 into the first required Amount
- Choose a Tax. In this case it is HST = 13%.
- The system will calculate that you paid $13.00 in Tax to make the total $113.

21.4 Create Expense

You can create a new expense directly from the Expenses list page by clicking on the **Create** button located at the top right side of the page. A modal will open for you to enter expense information.

- **Amount**: The amount of the expense.
- **Category**: Select the category from the drop-down list.
- **Vendor**: If you have already saved a vendor, you'll only need to start typing and it will auto complete.
- **Currency**: Select the currency of the expense.
- **Tax 1**: Enter tax 1 as percentage.
- **Tax 2**: Enter tax 2 as percentage
- **Notes**: Enter a description of the expense. When the expense is converted to an invoice, the text you enter here will feature as the line item description for the expense on the invoice.
- **Project**: Select a project attached to the expense
- **Expense Date**: Date when the expense incurred
- **Recurring**: Whether the expense should recur
- **Billable**: Whether the expense is billable
- **Show to Client**: Hidden or visible to client

- **Billed**: Whether the expense is already billed
- **Tags**: Custom tags e.g internet,rent etc
- **Receipts**: Upload expense receipts

21.5 Recurring Expenses

Some expenses are incurred on a consistent basis over a period of time, and manually recording them each time can get really tedious. Generating these expenses can be automated in Workice, resulting in systematic tracking.

To make an expense recur, edit the expense and modify **Recur Every** field.

You can set a repeat interval such as `daily, week, month, quarter, six months, year`.

Once you have chosen the interval, you can decide for the repeat to stop by a particular date.

Tip: To stop a recurring expense, edit the expense and change **Recur Every** field to **None**.

21.6 Invoice Expense

Are you billing a client directly for an expense? To bill an expense, first create an invoice for the client. Open the new invoice and just below the Client address, you'll see a red button **Expenses Available** click on the button to open the modal where you can select the expenses to include in the invoice.

CHAPTER 22

Calendar

Workice CRM Calendar module allows users to easily schedule, view, and manage their activities like calls, meetings and appointments, all in one place. Easily keep track of all your customer invoices, payments, expenses, leads, tasks etc

22.1 Add Calendar Type

By default 2 calendar types are available, `Work and Personal`. You are free to add/update or delete calendar types by clicking on the **Gear** icon at the top right of the Calendar page.

22.2 iCal Feed URL

Click on iCal buton to see options for pulling events from your Workice Calendar and pushing them into other calendar apps. You will now see a popup with your iCal Feed URL. Follow the on-screen instructions to "subscribe" to events in your Workice Calendar with other apps. New events created in Workice CRM will then show up automatically in those other apps.

22.3 Google Calendar

To display events from your Google Calendar, Go to **Settings** -> **System Settings** and enter your Google Calendar API key and your Google Calendar ID. Once the settings are configured, your events will display on Workice calendar.

CHAPTER 23

Tickets

Tickets are the means through which your end users (customers) communicate with agents in your support system. Workice support system allows you to present a web portal to your customers to create, track, and respond to support requests. Regardless of the type of customer support you provide, the one constant for all support organizations is that customers seek you out to help them resolve their issues.

Here are some of the options that your customers have for contacting you:

- Send an email
- Fill out a support request form in your Workice Support portal
- Fill out a support request form on your own web site (Use Ticket API)

Note: By default, a comment you enter as a Public Reply will be public and visible to everyone who views this ticket, including the person who requested support. You can also add private comments (referred to as an Internal Note). The requester never sees these notes, they are used for internal communication only. For example, an agent may need to get advice from another agent to solve the requester's support issue.

If you want to add an internal note, start your comment with **[NOTE]**

Example internal note;

```
[NOTE] My internal note comment
```

23.1 Create Ticket

To create a ticket, click **Create** button on the ticket list page. You will be redirected to the create ticket page where you can enter ticket information.

- **Department**: The department a ticket is associated with

Note: Custom fields will be displayed based on the selected department.

- **Subject**: Ticket subject e.g Billing Issue
- **Reporter**: The ticket reporter
- **Project**: Select a project (if any) associated with the ticket.
- **Priority**: Ticket priority (low, medium, high)
- **Message**: Ticket body/message
- **Tags**: Custom tags e.g logos, design etc
- **Files**: Attach ticket files e.g screenshots

23.2 Convert Ticket to Task

To convert a ticket to task, open the ticket and click **Convert to task** button. A popup modal will appear where you can select a project you want the task created. Click **Ok** to complete the action.

23.3 Change Ticket Status

To change the status of a ticket including closing a ticket, open a ticket and click on **Status** button and select your preferred status from the dropdown list.

23.4 Bulk Actions

If you need to perform an action for a number of tickets, you can do it in one click with the bulk action feature. To use the bulk action feature, mark the relevant tickets in their checkbox at the far left of the tickets list. Once you've marked the tickets, select an action to perform on them in the buttons below the ticket list page.

- **Close**: Close selected tickets
- **Archive**: Archive selected tickets.
- **Delete**: Delete selected tickets.

23.5 Customer Satisfaction Ratings

If you enable Customer Satisfaction Ratings in Workice Support, your customers will receive an email notification after their ticket has been solved asking them to rate their support experience. You can modify the number of days before a rating request is sent in **Settings > Ticket Settings > Feedback Request**. Setting it to `zero` will disable the feature.

23.6 AnswerBot

Answer Bot works right alongside your support team to help answer your customers' questions. With content from your Workice knowledge base, Answer Bot suggests articles to your customers to resolve their issues. The customer reviews the articles and if an answer is found, they can mark their question as answered and close the ticket.

CHAPTER 24

Reports

You can only manage what you can measure. With Workice CRM, manage the way your business is run with reporting that breaks down key metrics relating to your sales trends, invoices, projects, expenses and more. Reports are a great way to not only see how you're doing, but also forecast where you're going in the future.

Attention: All the charts and statistics displayed in the reports section will use your base currency.

24.1 Invoices Report

Your Invoice Details Report gives you a comprehensive overview of all Invoices you have generated in a given time period. You can run an Invoice Report by doing the following:

- Click on the Reports tab
- Select Invoices module from the **Modules** dropdown menu.
- Click **Reports** button.

There are a few options you can change when generating reports;

- **Date Range** – You can set a date range you would like this report to include information within
- **Clients** – You can run Client(s) specific Invoice reports if necessary
- **Recurring** – Choose to only include recurring invoices
- **Status** – Choose to only include Invoices with specific statuses (such as Not Paid, Fully Paid, Cancelled etc)
- **Sent** – Only display sent invoices

24.2 Deals Report

These options are for simple and effective analysis of the bottom line: winning or losing deals.

These reports are excellent milestone reports, to demonstrate how many deals were closed in the account.

Note: If you have setup cron correctly, sales velocity and conversion rate will be calculated daily at 04:00.

24.2.1 Sales velocity

- Displays the average amount of time a deal's creation until the deal it marked as WON or LOST.
- Recommended for gaining insight into how effective your users or processes are in closing deals quickly.

24.2.2 Conversion Rate

- Displays how many deals are won compared to the total number of deals.
- Recommended for knowing if you lose many of your deals allowing you to make improvements to address any issues.

24.2.3 Deals won

- Displays the number of deals marked as WON in a particular month.
- Recommended for an easy at-hand report of how your business is bringing in income through its sales process.

24.2.4 Deals lost

- Displays the number of deals marked as LOST in a particular month.
- Recommended for an easy at-hand report of the deals that did not make it through, allowing you to do better in the future.

You can run Deals Report by doing the following:

- Click on the Reports tab
- Select Deals module from the **Modules** dropdown menu.
- Click **Reports** button.

There are a few options you can change when generating reports;

- **Date Range** – You can set a date range you would like this report to include information within
- **Pipeline** – You can run reports for a specific pipeline
- **Stage** – Choose to only include deals in specific stage
- **User** – Choose to only include deals assigned to a user
- **Status** – Only display deals in open, won or lost status

24.3 Leads Report

A CRM leads report shows you how many leads your sales team has acquired in a given time frame, as well as the lead source and stages. You can run leads report by doing the following:

- Click on the Reports tab
- Select Leads module from the **Modules** dropdown menu.
- Click **Reports** button.

There are a few options you can change when generating reports;

- **Date Range** – You can set a date range you would like this report to include information within
- **Stage** – Choose to only include leads in specific stage
- **Source** – You can run reports for a specific lead source e.g Google
- **Sales Agent** – Choose to only include leads assigned to a user
- **Archived** – Only display archived/unarchived leads

24.4 Expenses Report

The Expense Report breaks down all of your expenses into detail. You can run expenses report by doing the following:

- Click on the Reports tab
- Select Expenses module from the **Modules** dropdown menu.
- Click **Reports** button.

There are a few options you can change when generating reports;

- **Date Range** – You can set a date range you would like this report to include information within
- **Client** – Choose to show expenses for a specific client
- **Project** – You can run reports for a specific project
- **Billable** – Choose to only include billable expenses
- **Billed** – Only display billed/unbilled expenses
- **Category** – Only display expenses in a specific category

24.5 Payments Report

Your Payments Report breaks out all of the payments you have recorded in Workice. You can run payments report by doing the following:

- Click on the Reports tab
- Select Payments module from the **Modules** dropdown menu.
- Click **Reports** button.

There are a few options you can change when generating reports;

- **Date Range** – You can set a date range you would like this report to include information within

- **Project** – You can run reports for a specific project
- **Client** – Choose to show expenses for a specific client
- **Invoice** – Choose to only include payments for an invoice
- **Payment Method** – Only display payments received in a specific payment provider

24.6 Estimates Report

The estimates report shows you how many estimates have been sent, approved or rejected in a given time frame. You can run estimates report by doing the following:

- Click on the Reports tab
- Select Estimates module from the **Modules** dropdown menu.
- Click **Reports** button.

There are a few options you can change when generating reports;

- **Date Range** – You can set a date range you would like this report to include information within
- **Client** – Choose to show estimates for a specific client
- **Status** – Choose to show accepted, declined or pending estimates
- **Sent** – Include only those estimates that have been sent/not sent
- **Invoiced** – Choose to only include invoiced or uninvoiced estimates

24.7 Credits Report

The creditnotes report shows you how many credits have been closed or open in a given time frame. You can run creditnotes report by doing the following:

- Click on the Reports tab
- Select Creditnotes module from the **Modules** dropdown menu.
- Click **Reports** button.

There are a few options you can change when generating reports;

- **Date Range** – You can set a date range you would like this report to include information within
- **Client** – Choose to show credits for a specific client
- **Status** – Choose to show open or closed credits
- **Sent** – Include only those credits that have been sent or not sent

24.8 Projects Report

Within Workice Projects, you can generate a project report, designed to give you an overview of all projects within your CRM. You can run projects report by doing the following:

- Click on the Reports tab
- Select Projects module from the **Modules** dropdown menu.

- Click **Reports** button.

There are a few options you can change when generating reports;

- **Date Range** – You can set a date range you would like this report to include information within
- **Client** – Choose to show projects for a specific client
- **Status** – Choose to show active, done and on hold projects
- **With Contract** – Include only those projects which has a contract or does not have a contract

24.9 Tasks Report

The Tasks Report lists all tasks in your CRM, and allows you to filter by stage, milestone and task project. You can run tasks report by doing the following:

- Click on the Reports tab
- Select Tasks module from the **Modules** dropdown menu.
- Click **Reports** button.

There are a few options you can change when generating reports;

- **Date Range** – You can set a date range you would like this report to include information within
- **Project** – Choose to show tasks for a specific project
- **Milestone** – Choose to show tasks linked to a specific milestone
- **Stage** – Include only those tasks in a particular stage
- **User** – Include only those tasks created by a particular user

24.10 Timesheet Reports

The Timesheet Report is a very helpful way for your business to review how much time is being spent on a given project and by whom. Time Reports show the hours and billable information for each project, task and team member.

You can run time report by doing the following:

- Click on the Reports tab
- Select Timesheets module from the **Modules** dropdown menu.
- Click **Reports** button.

There are a few options you can change when generating reports;

- **Date Range** – You can set a date range you would like this report to include information within
- **Project** – Choose to show time entries for a specific project
- **Task** – Choose to show time entries for a specific task
- **Billable** – Include only those time entries that are billable or unbillable
- **User** – Include only those time entries created by a particular user

24.11 Ticket Reports

The tickets report show the number of tickets created and solved by date.

You can run tickets report by doing the following:

- Click on the Reports tab
- Select Tickets module from the **Modules** dropdown menu.
- Click **Reports** button.

There are a few options you can change when generating reports;

- **Date Range –** You can set a date range you would like this report to include information within
- **Reporter** – Choose to show only those ticket opened by a specific user
- **Status** – Choose to show tickets in specified status e.g open, closed etc
- **Department** – Choose to show tickets for a specific department
- **Priority** – Choose to show tickets that have low,high or medium priority

24.11.1 Agent Performance

You can view agent performance by clicking on **Type** button in the **Ticket Reports** page.

The agent performance table contains the columns below;

- **Name**: Ticket agent name
- **Comments**: The number of ticket comments posted by the agent
- **Resolved**: The number of tickets solved by the agent
- **Feedback**: The percentage number of happy customer feedbacks

24.11.2 Ticket Feedbacks

You can view ticket ratings by clicking on **Type** button in the **Ticket Reports** page.

The ticket feedback table contains the columns below;

- **Name**: Ticket requester name
- **Ticket**: The ticket that the ticket requester rated
- **Date**: Date when the rating was received
- **Rating**: Whether the client was happy/dissatisfied

CHAPTER 25

System Settings

You can view and edit the system configuration in your CRM by going to **Settings** and clicking on the **System Settings**.

You can modify the settings below;

- **Purchase Code**: Enter your purchase code (Used for license verification)
- **Default Language**: The default system language
- **Locale**: Your preferred system locale
- **Timezone**: Your timezone
- **Default Currency**: Your preferred currency. Default USD
- **Currency Symbol**: Must be the symbol of your default currency above
- **Default Calendar**: System default calendar type e.g Work or Personal
- **Tax 1**: Default Tax 1 percentage
- **Default Subscription**: Enter your default subscription name
- **Tax 1 Label**: Give your Tax 1 a name e.g GST
- **Tax 2**: Enter the default Tax 2 percentage
- **Tax 2 Label**: Give your Tax 2 a name
- **Tax Decimals**: The number of decimals to be used in tax display
- **Quantity Decimals**: Number of decimals to be used in displaying quantity
- **Date Format**: Choose your preferred date format
- **File max size**: Maximum file size in KB e.g 8192 for 8MB
- **Allowed files**: Comma separated list of allowed files e.g png,pdf,docx etc
- **Privacy Policy URL**: A link to your privacy policy
- **Slack webhook URL**: Your company slack webhook url to receive notifications
- **Wablas Token**: API Key obtained from Wablas

- **WhatsApp Number**: Default WhatsApp number
- **WhatsApp Subscribe Text**: Text that users need to send to your whatsapp to subscribe. Default is SUB
- **Open Exchange API Key**: If left blank, the system will use our rates server to fetch currency rates.
- **Google Calendar API Key**: Your Google Calendar API Key
- **Google Calendar ID**: Your Google Calendar ID. It will look something like "abcd1234@group.calendar.google.com".
- **Default Role**: When a user is created or registered, this role will be assigned.

> **Attention:** Do NOT set the **Default Role** to admin.

Other Options

- **Auto Reminder**: Send email reminders for estimates, contracts, tasks or todos when almost overdue.
- **Enable Languages**: When enabled, you can select your preferred language using the dropdown languages menu.
- **Use Gravatar**: Use gravatar to retrieve user avatars
- **Allow Client Registration**: Enable client registrations
- **Show amount in words**: Show amount in words (Invoices, Credits, Estimates and Payments)
- **Daily Digest**: Enable this option to receive daily email digest
- **Exchange Rates**: Enable exchange rate updates
- **Use ReCaptcha**: Use ReCaptcha in registrations
- **Clients add projects**: Clients can add projects
- **Stop timer logout**: When this option is enabled, running timers will be stopped when you logout.
- **Contract to Project**: Automatically convert a contract into a project once it's signed
- **Update Notifications**: Receive update notifications
- **WhatsApp Enabled**: Enable/Disable WhatsApp integration.

CHAPTER 26

Lead Settings

You can view and edit the lead configuration in your CRM by going to **Settings** and clicking on the **Lead Settings**.

You can modify the settings below;

- **Default Stage**: The default lead stage
- **Default Sales Agent**: The default user assigned to leads
- **Enable double opt-in**: Enable this option to request lead consent
- **Auto Delete**: When a lead is converted to a customer, delete it.

26.1 IMAP Settings

Workice keeps track of all your emails, no matter where you send or receive them. When this option is enabled, email conversation history with each lead will be pulled in and become instantly visible in your lead email conversations. Sending an email to multiple recipients? We will notify you when a lead opens an email and at what time.

- **IMAP**: Enable IMAP email retrival
- **IMAP Host**: Your IMAP Host e.g imap.gmail.com
- **IMAP Username**: Your IMAP username/full address
- **IMAP Password**: Imap Password
- **Mail Port**: IMAP port e.g 587
- **Mail Flags**: IMAP flags. Default /imap/ssl/novalidate-cert
- **Mailbox**: Mail folder. Default INBOX

CHAPTER 27

Deal Settings

You can view and edit the deal configuration in your CRM by going to **Settings** and clicking on the **Deal Settings**.

You can modify the settings below;

- **Deal Rotting**: Number of days before a deal is marked as idle
- **Auto Invoice**: Automatically create an Invoice when deal is won
- **Default Pipeline**: The default deal pipeline
- **Default Stage**: The default deal stage. `Ensure the stage is in the default pipeline above`
- **Default Deal Owner**: The default user assigned to deals

27.1 IMAP Settings

We offer two easy and time-saving ways to track email conversations. In addition, we'll connect your emails automatically to your contacts and deals, so you'll have a full overview of everything that's happening with a prospect in one place. You can now save time and improve your sales productivity.

- **IMAP**: Enable IMAP email retrival
- **IMAP Host**: Your IMAP Host e.g imap.gmail.com
- **IMAP Username**: Your IMAP username/full address
- **IMAP Password**: Imap Password
- **Mail Port**: IMAP port e.g 587
- **Mail Flags**: IMAP flags. Default /imap/ssl/novalidate-cert
- **Mailbox**: Mail folder. Default INBOX

CHAPTER 28

Invoice Settings

You can view and edit the invoice configuration in your CRM by going to **Settings** and clicking on the **Invoice Settings**.

Note: Any changes you make to the Invoice Settings will apply to all your invoices.

You can modify the settings below;

- **Invoice Color**: Your preferred invoice color (Company Color)
- **Creditnote Color**: Your preferred credit notes color
- **Invoice Prefix**: Add your chosen prefix. For example, you may choose to add your company initials, such as W&M and Invoices will appear as W&M0001 etc Default **INV**
- **Creditnote Prefix**: Add your chosen creditnote prefix e.g CN will appear as CN0001. Default **CN**
- **Invoice Number Format**: Invoice format. Default -[yyyy][mm][dd]-[i] for {year}{month}{day}-{number}
- **Credits Number Format**: Credit note format. Default -[yyyy][mm][dd]-[i] for {year}{month}{day}-{number}
- **Invoices due after**: Number of days before an invoice is overdue.
- **Reminder 1**: Number of days when a standard overdue reminder is set to send. Default 1
- **Reminder 2**: Number of days when a second reminder should be sent after invoice overdue. Default 5
- **Reminder 3**: Number of days when a late fee should be applied and final reminder sent. Default 10
- **Late Fee on Last Reminder**: Apply late fee to invoice on 3rd reminder
- **Late Fee**: Late fee percentage
- **Starting Number**: The number to start counting your invoices from e.g 100 will start from INV00100
- **Swap Company/Client Address**: Swap Company and customer addresses left/right
- **Display PDF Badge**: Display PDF Badges (partially implemented)

- **Email on Recur**: Automatically send recurred emails when they recur
- **Thank You Emails**: Send Thank You emails when payments received
- **Show Item Tax**: Show Invoice item tax
- **Show Credits Tax**: Show Credits tax
- **Archive Invoice**: Automatically archive paid invoices
- **Apply Credits**: Automatically apply credits on recurring invoices (if any)
- **Invoice Logo**: Upload your Invoice Logo
- **Invoice Footer**: Any additional information that should appear at the bottom of your invoices
- **Creditnote Footer**: Any additional information that should appear at the bottom of your credits
- **Default Terms**: Default invoice terms.
- **Creditnote Terms**: Default credits terms

CHAPTER 29

Payment Settings

You can view and edit the payment configuration in your CRM by going to **Settings** and clicking on the **Payment Settings**.

Note: Supported payment gateways are **paypal, stripe, mollie, razorpay, checkout, braintree, bank, wepay**

You can modify the settings as below;

- **Payment Prefix**: Add your chosen transactions prefix. Default PAY
- **Payment Number Format**: Estimate format. Default -[yyyy][mm][dd]-[i] for {year}{month}{day}-{number}
- **Payment Gateways**: Enabled payment gateways separated by a comma
- **Paypal Live**: Enabled live paypal payments or sandbox
- **Paypal Email**: Your Paypal email address for receiving payments
- **2checkout Live**: Enable Live/Sandbox payments via 2Checkout
- **Braintree Live**: Enable Live/Sandbox payments via Braintree
- **Braintree Merchant Account**: Default Braintree merchant account
- **Wepay Live**: Enable Live/Sandbox payments via WePay
- **Bank Details**: Your Bank information to be displayed to client incase they need to pay via Bank

29.1 Adding Custom Gateway

Workice allows you to integrate with 3rd-party payment gateways or services. The following steps summarize how to develop a custom payment gateway:

- Create a payment gateway form that allows customers to enter any required payment data.
- Create a custom payment gateway provider class that performs the required payment processing.

- Map the payment gateway provider to the appropriate payment form by registering a custom implementation of the PaymentInterface interface.
- (Optional) Create an IPN handler for your payment gateway.
- Open **Settings > Payment Settings** and add the name of your custom gateway to the list of comma separated Payment Gateways.
- Customers on your portal can now select the custom payment method during checkout and pay for their invoices using the given gateway.

29.1.1 Creating payment gateway forms

- Start by creating a file named **form.blade.php** in `Modules\Payments\Resources\views\{gateway}\form.blade.php` where `{gateway}` is the name of your custom gateway.
- We recommend you to take a look at the other gateways files in order to get the idea. The form submits to a custom controller (we'll create it later).
- Create a route in `Modules\Payments\Http\custom.php` (There is a sample route in the file). Your form should use this route to submit data to the controller.
- Inside your custom payment gateway **form.blade.php** before closing your form add this line **@include('payments::_includes._options')** this will show options to the client i.e Pay Full amount or partial amount.
- Create a Controller in `Modules\Payments\Http\Controllers` e.g `ExampleGatewayController`.
- Inside this controller handle checkout request when the form is submitted. Take a look at `Modules\Payments\Http\Controllers\Base` controllers for an idea on how to implement your checkout method.
- After initializing and processing the payment in controller checkout method, now verify the payment and post the data to `PaymentEngine` helper as shown below;

```
// Check if payment successfull
if ($charge->paid == true) {
 // Pass the data to PaymentEngine class as shown below
 // Replace {gateway} with your gateway name
 $payment = (new \Modules\Payments\Helpers\PaymentEngine('{gateway}', $charge))->
↪transact();
}
```

Now create your payment gateway class in `Modules\Payments\Gateways` to parse data and save to Workice CRM. If your custom gateway is named **{example}** ensure this file is named **Example.php**. This class must extend `Modules\Payments\Contracts\PaymentInterface` interface. Take a look at how the other payment gateways are parsing data and submits it to `Modules\Payments\Helpers\Cashier` class.

```
public function pay($transaction)
{
 // Get the invoice from the database
 $this->invoice = $this->invoice->findOrFail($transaction->orderId);
 // The getData() method returns formatted array of data to be passed to Cashier␣
↪class.
  $data = $this->getData($transaction);
 // Pass the transaction data to Cashier as first parameter
 // The second parameter is the invoice object
```

(continues on next page)

(continued from previous page)

```
  return (new Cashier($data, $this->invoice))->save();
}
```

29.1.2 Handling IPN for custom payment gateways

If you wish to use Instant Payment Notification (IPN) with your payment gateway, you need to create a custom HTTP handler that performs the required communication between your Workice application and the payment service. Do not handle IPN requests using standard pages (web forms).

We recommend creating IPN controller in `Modules\Webhook\Http\Controllers` and add a route in `Modules\Webhook\Http\routes.php`. Implement your IPN verification and send the data to the PaymentEngine as shown below;

```
// {gateway} is the name of your gateway and {data} is the transaction data
$txn = (new \Modules\Payments\Helpers\PaymentEngine('{gateway}', {$data}))->
↪transact();
```

> **Attention:** To enable your custom gateway, add it to the list of active payment gateways in **Settings > Payment Settings > Payment Methods**

CHAPTER 30

Estimate Settings

You can view and edit the estimate configuration in your CRM by going to **Settings** and clicking on the **Estimate Settings**.

Note: Any changes you make to the Estimate Settings will apply to all your estimates.

You can modify the settings below;

- **Estimate Color**: Your preferred invoice color (Company Color)
- **Estimate Prefix**: Add your chosen prefix. For example, you may choose to add your company initials, such as W&M and Estimates will appear as W&M0001 etc
- **Estimate Number Format**: Estimate format. Default -[yyyy][mm][dd]-[i] for {year}{month}{day}-{number}
- **Estimate Starting Number**: The number to start counting your estimates from e.g 100 will start from EST00100
- **Estimate Footer**: Any additional information that should appear at the bottom of your estimates
- **Estimate Terms**: Default estimate terms
- **Display PDF Badge**: Display PDF Badges (partially implemented)
- **Show Item Tax**: Show Estimate item tax
- **Estimate to Project**: Turn estimates to projects once approved
- **Estimate to Invoice**: Automatically turn an estimate to invoice when approved by client
- **Archive Estimate**: Automatically archive converted estimates

CHAPTER 31

Ticket Settings

You can view and edit ticket configuration in your CRM by going to **Settings** and clicking on the **Ticket Settings**.

You can modify the settings below;

- **Ticket Prefix**: The default ticket prefix e.g TKT
- **Ticket Number Format**: Ticket code format e.g -[yyyy][mm][dd]-[i] (yyyy represents the Year, mm for month, dd for day).
- **Ticket Start Number**: Ticket code numbers will start from the value entered here.
- **Default Department**: When a department is not specified when creating a ticket, this department will be used.
- **Auto Close Ticket**: Closes tickets that have not been active for X days
- **Ticket Due Days**: The default number of days a ticket should take before it's closed. Default 3 days
- **Feedback Request**: Ask for closed ticket review after X days. Enter 0 to disable
- **Enable Answer Bot**: Auto search the knowledgebase and attempt to find articles that match what the customer is looking for. Answerbot will send an email with a link to ticket related articles.

31.1 IMAP Settings

Integrate with your existing email infrastructure to receive tickets via email. For those customers that prefer to use email, Workice provides powerful email integration functionality that can intelligently read incoming emails and generate new help desk tickets as well as add comments to existing open tickets.

- **IMAP**: Enable ticket email retrival via IMAP
- **IMAP Host**: Your IMAP Host e.g imap.gmail.com
- **IMAP Username**: Your IMAP username/full address
- **IMAP Password**: Imap Password
- **Mail Port**: IMAP port e.g 587

- **Mail Flags**: IMAP flags. Default /imap/ssl/novalidate-cert
- **Mailbox**: Mail folder. Default INBOX

CHAPTER 32

Email Templates

You can modify email templates for invoice emails, estimates, projects etc. To do so, select **Settings** from the main menu sidebar then click **Translations**.

Click on **Emails** button at the top bar to open email translations.

Select the language you want to modify;

> **Attention:** The english version variables are named in {module}.{action}.{type} format (dot notation). Example; if you need to edit the message that will be sent when you send an invoice, look for a variable named `invoices.sending.body`. To edit sent message subject, modify `invoices.sending.subject` value.

32.1 Example

Let's say we want to modify the German message that should be sent to us when we receive a payment.

Go to Settings and click on Translations;

- Click **Emails** button.
- Look for German Language and click the **Pencil** icon to modify text
- Look for English translation `payments.received.body`.
- The English version displayed is `You have received payment of :amount on :date for invoice :code`
- Change the text that appears at the right side (text box) by replacing it with your German version.
- Next time you receive a payment notification message in German, your custom message will be displayed in the email body.
- Click **Save** button to save the changes

Note: You can modify other email templates in different locales using the same procedure above.

CHAPTER 33

Translations

Translations are a collection of visual elements in the Workice application, like labels, information messages, notifications, alerts, work flow statuses, etc

To modify the text translated to the target language, navigate to Settings > Translations in the main menu.

- Click on the **Pencil** icon of the language you want to modify.
- Click the **Pencil** icon to modify the language file i.e app or activity language files
- Replace the text displayed on your right side with your language equivalent
- English version is displayed on the left side.

33.1 Adding Translations

If you want to add translations to Workice;

- Create a folder for your language example `ms`
- Copy all language files from folder /resources/lang/en into your new folder `ms`
- After copying the translation files you can now modify the language lines in Settings > Translations

Alternatively, you can share your translations here and we'll be glad to add it to workice.

33.2 Backup Translations

To backup your custom translations.

- Navigate to **Settings** > **Translations** and click on **Backup** button.
- A JSON file will be downloaded with your custom translations

33.3 Restore Backup Translations

To restore your translations.

- Navigate to **Settings** > **Translations** and click on **Restore** button.
- Upload your backup file to restore your translations

Attention: This action will overwrite your current translations

Note: If you need your translations added to next updates you can help translate it on Github