

# Kimai Time-Tracking documentation

# Table of Contents

## Introduction

### 1. Installation

#### 1.1 Updates

### 2. Using Kimai

### 3. Developer

#### 3.1 Introduction

#### 3.2 Translations

#### 3.3 Invoice templates

#### 3.4 Documentation

### 4. FAQ

# Kimai Time-Tracking documentation



Your Time.



Welcome to our short documentation ... which is unfortunately not very comprehensive yet.

We could use your help improving the documentation, [please contact us in the forum](#) if you want to help or read the Developer/Documentation section.

## Download for offline usage

There are some downloads available for Kimai documentation:

- [This document as PDF](#)

There is also a german version, contributed by a Kimai user:

- [German manual as DOCX](#)
- [German manual as ODT](#)
- [German manual as PDF](#)

# Installation

# Installation

The following requirements must be met on the server installation:

- PHP version 5.2 or above
- MySQL libraries for PHP
- one MySQL database (as Kimai supports table prefixes, this database may be shared with other tools)
- FTP or shell access to set the correct filesystem permissions.

Extract the Kimai release zip/tarball into a server directory accessible from the outside and navigate to this location using a modern Webbrowser. A step-by-step installation program is included with every new release of Kimai. Follow the instructions of the installer and correct missing or incorrect permissions using your FTP or shell account. If the step-by-step installer does not work, please report this error using the [support forum](#).

Remove the “install” directory after the installation!

After you removed the “install” directory you may login into Kimai using the following credentials:

- user: admin
- password: changeme

Set the administrator password using the setting dialog. Keeping “changeme” as your password is a security hole!

## Updates

### Step 1

To install a new version just replace the entire directory with the new package EXCEPT the file `autoconf.php` which can be found inside the `/includes` folder. Additionally you have to set writing permissions to all `../compile` directories (don't forget the ones in the subfolders of your extensions!) so that PHP can write to it. Writing permission must also be granted to the `/temporary` folder and the included `/temporary/logfile.txt`.

Attention: Kimai will NOT work with wrong writing permissions!

### Step 2

Finally visit your Kimai installation with your web browser and the auto update process will perform all further necessary updates.

A new version updates the database automatically (is necessary). Before that a backup of your data will be created. Normally you'll not even notice this process. If anything goes wrong during the database-update you can recover the data via phpMyAdmin (we're also working on a tool for restoring...).

**CAUTION:** This software is free of charge. That doesn't mean, that you are allowed to remove copyright-notes (Neither on the login-panel nor in the credits-window). Of course you're not allowed to remove the link to the credits... If those notes are offending you and you do not agree with these rules - DON'T use Kimai.

# Using Kimai

# Using Kimai

This is the Kimai manual - our end-user documentation. Its english version is not very comprehensive, we could use your help in improving it. [Please contact us in the forum](#).

## Extensions

The timesheet, export or expenses screen which you get by clicking the corresponding tab is called an extension. To enable Extensions just delete the hash (#) symbol in front of the folder name. To disable, do it otherwise.

For developing an extension please refer to the [Kimai forum][2].

## Timesheet

Here you can track your time. The surface is easy to understand and use.

### Tracking your time

Just select a customer and all corresponding projects will be highlighted. Next step is to choose the project you are working on and last but not least the task you are doing. Now you are ready to track your time. Just hit the big green button in the top right corner to start tracking. To stop your current record hit the red stop-button either on the left of your current record or the top right corner. Afterwards you can choose another customer, project and/or task and continue to record.

### Editing your time

If you have forgot to quit tracking or selected a wrong task: don't panic! There is a little pencil next to your recorded events. By clicking on it you will be able to rearrange a mistake that might have happened.

Attention: The start and end-time of events that have been recorded after that mistake will not be modified automatically. Have a look if there are overlapping entries.

If you are in need of some extra info to your current task use the comments bubble displayed beneath the running time. Here you can add whatever info you think can help you with explained what you did.

## Admin Panel

In the admin panel you can do every administration that is necessary for the running system.

### Customers

Here you can add, modify and delete customers. If you do not set a password the customer cannot login to Kimai. By selecting a group you enable all members of that group to see that customer. All fields except the customer name are optionally.

### Projects

Here you can add, modify and delete projects. Projects have to be aligned to a customer. Further you can set rates for this project if they differ from the standard rates (for details about rates see chapter 7.5). If there is a budget it can be set here, too.

### Tasks

Here you can add, modify and delete tasks. A task has no relation to a customer or a project, but you can select which groups can use them. Additionally you can set rates here, too (for details about rates see chapter 7.5).

### Users

Here you can add, modify and delete users. A user can only belong to one group at a time. If you enter a valid email

address you can send emails from your local account by clicking the mailer symbol.

## Groups

Here you can add, modify and delete groups and also define their leaders. Every group can have an unlimited amount of leaders.

## Advanced

Here you can define general settings like number of failed login-tries, bantime, currency, etc.

## Budget

This extension simply shows you a graphical overview over the budget for your projects.

## Expenses

Here you can add expenses to a project which, of course, affect the projects' budget.

## Export

With this extension you can export tracked time per User into an Excel-file, csv-file or a pdf-file. As a regular user you can only see your time. If you are at least a group leader you can select members of your group and export their data for reporting.

## Invoice

Here you can create invoices for your customers. Does not work properly, needs PDO connection to database. (03/05/2010)

## Database

The database will be created automatically by the installer. It contains the following tables:

Maintenance of the database can be done via Kimai. If there is still some rubbish left it can be identified by the trash value.

## Rates

If you have a look into the table kimai\_rates you may see something like this:

User	Project	Event	Rate
1	2	5	10.00
2	NULL	5	20.00
2	2	NULL	30.00
2	NULL	NULL	40.00
NULL	2	5	50.00
NULL	NULL	5	60.00
NULL	2	NULL	70.00

The entries are already sorted. Kimai takes the first row that exists (maybe the one with the most hits).

Explanation of the table:

- user 1 on project 2 and task 5 has a rate of 10.00 € per hour
- user 2 on task 5 has a rate of 20.00 € per hour
- user 2 on project has a rate of 30.00 € per hour
- user 2 has a rate of 40.00 € per hour



- project 2 and task 5 have a rate of 50.00 € per hour
- task 5 has a rate of 60.00 € per hour
- project 2 has a rate of 70.00 € per hour

Some examples:

- User 3 is working on project 2: his rate is 70.00 €
- User 2 is working on project 1 and task 5: his rate is 20.00 €
- User 2 is working on any project except the defined ones for him: his rate is 40.00 €

This table can only be edited with SQL statements or a database-tool (05/03/2010).

Developer

# Developer

There are several ways to get a copy of Kimai.

## Releases

From time to time we will release ZIP archives. These are most often beta versions, to be tested by our user base for problems and bugs. Production ready releases will be linked specifically in the download section at our website.

You can download the current developer-version as ZIP from our [master branch at GitHub](#).

## From The Repository

To get the developer version use the repository. Please keep in mind, that this version is NOT stable! We like to receive bug reports to be faster with fixing them, but we can not guarantee support for these versions.

We recommend using a fork, so any changes made can be pushed back and then submitted back via pull-requests:

- Fork Kimai
- Clone it locally: `git clone https://github.com/YOUR-USERNAME/kimai.git`

Then, if you want to make changes and submit them, use a feature branch:

- `git checkout -b my-feature`
- `git commit -am "Added my feature"`
- `git push origin my_kimai`

Create a pull request from your GitHub repo page.

## Build Kimai from source

Nightly Builds can be created by zipping the "core"-folder. Thats all you have todo.

For beta versions we have to edit the file "includes/version.php" and change `$kga['status']` to "beta".

For a release we have to do the following:

1. Deactivate "Debug Extension" by prepending a # to the folder name
2. Change `$kga['status']` to an empty string
3. Add `error_reporting(E_ERROR);` to "basics.php"

## Introduction

This document gives an overview of how Kimai works. It is intended for developers (core and extensions).

## Glossary

First there are some special words used in Kimai to describe things:

Floater (or Dialog)

Those are the draggable “windows” which open as an overlay in the page. For example the preferences dialog or the dialog to add new timerecords.

Sublists

The lists in the lower part which show users, customers, projects and events.

Processor

PHP files which handle AJAX calls.

Extension

Every tab you see in Kimai (like timesheet, export, admin panel) is an extension.

Database Layer

For every type of database connection (currently MySQL and PDO) a so called database layer is used. This is essentially a file which provides function to access the database.

## Folder structure

- **compile** SMARTY places the compiled template files from the core here
- **core**
- **css** stylesheets which are independant of any skin
- **extensions** every subfolders is one extension, for details see the chapter below
- **grfx** graphics which are independant of any skin (like the logo)
- **includes** Files which are needed in almost every other file.
  - **autoconf.php** Contains the configuration given by the user during setup. (Database connection information)
  - **basics.php** Simple wrapper to include all usually necessary files like functions and database layer.
  - **db\_layer\_mysql.php** as an example this is the database layer when the mysql extension is used to connect to the database
  - **func.php** Database independant functions are placed here.
  - **kspi.php** “Kimai Standard Processor Initialization” is included by every processor. Some setup is done here.
- **installer** Installer for Kimai.
- **js** Contains javascript files for the core.
- **languages** Every file is shown as a possible language (without the .php file extension) in the preferences dialog.
- **libraries** External libraries which are used (and possibly modified) by Kimai.
  - **jQuery** Some JQuery files, like datepicker.
  - **smarty** The template engine used by Kimai.
  - **tcpdf** Library to create PDF files using pure PHP.
- **skins** Currently contains only one skin. Other folders will be shown as available skins. Only the styles.css is mandatory. But for the sake of readability you should split your stylesheets up into several files.
- **templates** Templates used by the core, e.g. login forms and sublists.
- **temporary** Used for temporary file creation.

## Login Process

Login happens at the /index.php file. This is where you land when you open the kimai folder in your browser. Both, customers and users can login through the same login mask. For conflicting names (user and customer have the same name) the customer account takes precedence.

After a successfull login the user is redirected to /core/kimai.php. Here all extensions are loaded and the main user interface is being build.

## Extensions

When you develop an extension you have to think of a name and a so called key. This key should be prepended to every javascript method, class, id, ... just anything that is used within your extension to prevent conflicting names.

Every extensions needs a compile folder. This is where the smarty template engine stores the files which are passed to the client based on the template files from the templates folder. Floaters get their separate folder under the templates folders. The folder css is obviously for CSS files which are needed by the extension. The js folder usually contains two javascript files. An `_init.js` and a `_func.js` file. The first should contain code to setup the extension. The second should only contains javascript functions which are used within this extension.

Only after an extension/tab is selected its content is being loaded. This is handled by the `init.php` file in the extensions folder. The `processor.php` file handles AJAX calls and `floaters.php` creates those draggable windows.

## Translations

Kimai is continually evolving so translations get outdated pretty fast. Therefore we're always happy if people invest their time into improving the translations. Translations to new languages are welcome as well, of course.

### Where To Start?

The language folder in Kimai is where all translation files are stored. All files in that folder, which end in `.php` are displayed to the user as possible translations. For now we use a short code to identify its language.

### Format Of The Translation Files

The files are simple PHP scripts which return an array that in turn can contain arrays and that maps translation keys to their respective translation in the language. So if you have

```
'ext_invoice' => array( "invoiceTitle" => "Create invoice", "foo" => "Hello World" ),
```

only "Create invoice" is what is shown to the user and has to be translated. The strings "ext\_invoice" and "invoiceTitle" are the keys which Kimai uses to find the translated string.

### Adding New Translations

Doing so is pretty easy to describe, but it is some work for you. Just take the `en.php` file as a basis. This file contains all strings which can be translated in Kimai. Save that file under a new name (e.g. `nl.php`) and start translating the values (see above).

### Updating A Translation

If you want to improve an existing translation there are mainly two things you will do. Either you will change existing translated string (which surely doesn't need any explanation) or you will add missing translations. To do so check where the keys of the PHP arrays differ from the reference file `en.php`. If you find any keys missing in `en.php` you can remove them in your language file.

### Publishing A Translation

You can publish your finished translation file at: <http://forum.kimai.org/index.php?board=11.0>

Or even better, create a pull request and we include it in the official release: <https://github.com/kimai/kimai/pulls>

# Invoice templates

Creating templates for the invoice extension is actually pretty easy.

Just create an odt (Open Office Text Document) file and write the variables in the form of [var.variablename] (the brackets are required).

Variables you can use:

- [var.project] Name of the project
- [var.projectComment] Comment of the project
- [var.customerName] Name of the customer
- [var.companyName] Company of the customer
- [var.customerStreet] Street of the customer
- [var.customerCity] City of the customer
- [var.customerZip] Zip code of the customers city
- [var.customerComment] Comment, which is set for the customer
- [var.customerPhone] Phone number of the customer
- [var.customerFax] Fax number of the customer
- [var.customerMobile] Mobile number of the customer
- [var.customerEmail] Email address of the customer
- [var.customerContact] Name of the customer's contact person
- [var.customerVat] VAT of the customer
- [var.beginDate] Selected starting date
- [var.endDate] Selected end date
- [var.invoiceID] Defined as "CUSTOMER-NAME - YEAR - MONTH". See /extensions/ki\_invoice/print.php.
- [var.today] Current Date
- [var.dueDate] Defined as Current Date %2B 1 Month. See /extensions/ki\_invoice/print.php.
- [var.total] Total without VAT
- [var.gtotal] Total including VAT
- [var.ttltime] Total amount of hours

A row has these variables:

- [row.desc] Name of the task (event)
- [row.hour] Duration of the timesheet entry in decimal hours
- [row.fduration] Duration of the timesheet entry formatted as "hours:minutes"
- [row.amount] Wage of the entry
- [row.date] Day of the entry
- [row.description] Description of the timesheet entry
- [row.comment] Comment of the timesheet entry

If you get a blank page or an error similar to the follow

```
Fatal error: Uncaught exception 'tinyDocException' with message 'the PHP global variable named 'c<text:span' does not exist or is not set yet.' in /var/www/kimai/extensions/ki_invoice/TinyButStrong/tinyDoc.class.php:997 Stack trace:
#0 /var/www/kimai/extensions/ki_invoice/TinyButStrong/tinyButStrong.class.php(1870): tinyDoc->meth_Misc_Alert(Object(clsTbsLocator), 'the PHP global ...', true)
#1 /var/www/kimai/extensions/ki_invoice/TinyButStrong/tinyButStrong.class.php(669): clsTinyButStrong->meth_Merge_AutoVar('<?xml version="...', true)
#2 /var/www/kimai/extensions/ki_invoice/TinyButStrong/tinyDoc.class.php(269): clsTinyButStrong->Show(0)
#3 /var/www/kimai/extensions/ki_invoice/print.php(204): tinyDoc->saveXml()
#4 {main} thrown in /var/www/kimai/extensions/ki_invoice/TinyButStrong/tinyDoc.class.php on line 997
```

the variable name has different style applied. Select the (or all) variable name(s) and reset them to the default

format. Then you can format the whole variable name as you want it.

## Writing documentation

If you find some quirks in the documentation or found something worth mentioning, please don't hesitate to extend the documentation and send in pull requests.

Our repository for the documentation is located at:

- <https://github.com/kimai/documentation>

We use Markdown to write the documentation and the great [GitbookIO](#) tools to write and publish it.

You can either use the [GitBook Editor](#) or write the Markdown directly in your favorite editor. If you are going to add Chapters or Articles manually, make sure to include them in the file SUMMARY.md.

If you want to convert HTML documentation to Markdown, you can for example use the online tool <http://heckyesmarkdown.com/>

# FAQ



# FAQ

Falls diese kurze FAQ Deine Frage nicht klären kann oder Du mehr wissen willst, dann komm einfach im [Support Forum](#) vorbei.

## Wann wird Kimai 1.0 endlich veröffentlicht?

Das Kimai Projekt wird von den Entwicklern in Ihrer Freizeit programmiert. Da dies kein kommerzielles Projekt ist, können wir der Weiterentwicklung leider nicht immer so viel Zeit widmen, wie wir es gerne würden. Die Version 1.0 ist auf unbestimmte Zeit verschoben. Besucht bitte hin und wieder den Blog - sobald wir etwas neues rausbringen (egal ob Beta oder Final) wird es dort veröffentlicht.

## Kann ich meine Zeitdaten irgendwie sichern?

Der einfachste Weg ein Backup der Kimai Datenbank anzufertigen ist die Nutzung eines MySQL Backup Tools. Da wären z.B. die Export Funktion von phpMyAdmin, das Tool MySQL Commander oder phpMyBackupPro.

## Wieso kann ich keine Zeiten eintragen, die vor dem Installationszeitpunkt liegen?

"...ich habe das Problem, das ich keine Zeiten nachtragen kann, die älter als dem Installationsdatum sind. Ich würde gerne die Daten seit Beginn 02/09 eingeben. Jedoch wird mir im Dateselector das Installationsdatum angezeigt, sodass ich nicht erkennen kann, ob der ältere Eintrag übernommen worden ist. In der Datenbank finde ich den Eintrag."

Der Tag, an dem der Kimai-Kalender beginnt, lässt sich per Konfigurationsdatei nachträglich einstellen. Öffnen Sie dazu die Datei [kimai]/include/vars.php und verändern Sie folgenden Teil:

```
$kga['calender_start'] = "";&nbsp; ---&gt;&nbsp;&nbsp;&nbsp; ``$kga['calender_start'] = "31/12/2008";
```

```
// here you can set a custom start day for the date-picker. // if this is not set the day of the users first day in the system will be taken // Format: ... = "DD/MM/YYYY"
```

## Ich kann nur einen Teil Der Zeiteinträge für einen Zeitraum sehen, aber nicht alle

Die Anzahl der angezeigten Datensätze ist standardmäßig auf 100 limitiert. Sie können diese begrenzung unter Einstellungen (im Headerbereich neben dem Logout-Button) ändern.

## Beim Versuch Kimai 0.7.x zu installieren bekomme ich immer nur diese Fehlermeldung (ähnlich):

```
Fatal error: Smarty error:
unable to write to $compile_dir '...../includes/smarty/templates_c'.
Be sure $compile_dir is writable by the web server user.
in /Users/th/Sites/kimai_0.7.11/includes/smarty/Smarty.class.php on line 1095
```

Die Lösung ist einfach: Im Ordner kimai/includes/smarty/ finden Sie einen Ordner namens 'templates\_c'. Setzen Sie diesen Ordner auf schreibbar. Außerdem bitte nicht vergessen die Datei 'logfile.txt' im Ordner temporary schreibbar zu machen, sonst gibt es im nächsten Schritt wieder einen Haufen Errors ;)

Eine andere Fehlermeldung, die bei der Installation auftauchen kann, ist diese:

```
Fatal error: Call to a member function display() on a non-object in
/home/.../includes/connect.php on line 23
```

Überprüfen Sie in diesem Fall, ob der ins Config-File Datenbank-Benutzer das Recht hat darauf zuzugreifen.

---

## Kann ich mich irgendwie am Projekt beteiligen?

Wenn Du interesse hast, Mitglied des Teams zu werden, melde Dich bitte zuerst im [Forum](#) an damit wir uns kennenlernen können. Dank GitHub kannst Du uns jederzeit Pull-Requests zukommen lassen, auch Team-Mitglieder arbeiten so. Wenn Du viel beiträgst, nehmen wir Dich gerne in die Team Liste der offiziellen Contributor auf :)

Wir suchen dringend jemanden, der eine gute Doku schreiben kann. Gute Englischkenntnisse sind Voraussetzung. Am liebsten native.

Du bist super in CSS und brauchst unbedingt etwas, womit Du Langeweile todschlagen kannst? [Melde Dich!](#) Wir freuen uns über Mitstreiter, die etwas von Frontend-Arbeiten verstehen.

---

## Kann ich meine Einträge irgendwie als Stundenzettel anzeigen/ausdrucken - z.B. "gruppiert nach Projekten und Tätigkeiten" oder "Alle Einträge für Kunde XYZ für die Zeit vom xx bis xx"?

In der aktuellen Beta-Version können die Einträge über die normale Benutzeroberfläche angezeigt werden, und es können PDFs erzeugt werden. Wenn Sie einen Zeitraum im oberen Teil des Fensters gewählt haben, werden ihnen in der Benutzeroberfläche nur Zeiten bzw. Summen für diesen Zeitraum angezeigt. Eine weitere Anzeigemöglichkeit ist die für bestimmte Tage (tragen Sie dafür einfach den gleichen Tag in 'von' und 'bis' feld ein).

Es gibt in der 'Friendly Hacks' Sektion bereits ein paar Extensions, die dieses Problem angehen.

Natürlich ist geplant, dass Kimai in Zukunft selbst Reports erstellen kann. Aber leider dauert die Entwicklung seine Zeit - gerade wenn man 'nebenbei' noch einen Vollzeit-Job hat ....

---

## Ich möchte Beta-Tester werden. Wo bekomme ich die aktuelle Kimai-Version?

Der Beta-Test der neuesten Entwickler Version läuft kontinuierlich, unser Team ist zu klein um alles alleine zu schaffen. Bei Interesse meldet Euch bitte im Forum.

Du kannst entweder die Online Demo verwenden (siehe Webseite) oder noch besser: Du clonest Dir den aktuellen Master Branch von GitHub und testest diesen.