

1.5.14 Invoice Ninja

Invoice Ninja is a free, open-source invoicing, expenses, and time-tracking application for freelancers and small businesses, offering features like custom-branded invoices, recurring billing, online payments via 40+ gateways, project and task management, and expense tracking

Invoice Ninja is by far the most comfortable invoice system that I have used so far for my small freelance business in the hospitality industry. It has more features that I will ever need but it makes invoicing employers just so streamlined and easy. Instead of manually making all your invoices, sending emails, keeping track of payments, due dates, manual payments and more. Invoice Ninja makes this all a smooth experience for both you and the person that needs to pay you.

- [01. Docker Installation](#)
- [02. Payment Integrations \(Mollie\)](#)
- [03. Localization](#)

01. Docker Installation

Alright, so if you have a small business or you are simply a freelancer in any field then Invoice Ninja might be for you. It is quite an easy install but you need to know a couple of gotcha's to get things rolling. I show you here my method that works for me, and you can just copy paste that. However, there are of course other workflows possible as well.

According to its official documentation

Docker Compose File

```
services:
  app:
    build:
      context: .
    image: invoiceninja/invoiceninja-debian:${TAG:-latest}
    restart: unless-stopped
    env_file:
      - ./env
    volumes:
      # - ./php/php.ini:/usr/local/etc/php/conf.d/invoiceninja.ini:ro
      # - ./php/php-fpm.conf:/usr/local/etc/php-fpm.d/invoiceninja.conf:ro
      # - ./supervisor/supervisord.conf:/etc/supervisor/conf.d/supervisord.conf:ro
      - app_public:/var/www/html/public
      - app_storage:/var/www/html/storage
    depends_on:
      mysql:
        condition: service_healthy
      redis:
        condition: service_healthy
  nginx:
    image: nginx:alpine
    restart: unless-stopped
    ports:
      - 8333:80
    volumes:
      - ./nginx:/etc/nginx/conf.d:ro
```

- app_public:/var/www/html/public:ro
- app_storage:/var/www/html/storage:ro

depends_on:

app:

condition: service_healthy

mysql:

image: mysql:8

restart: unless-stopped

environment:

MYSQL_DATABASE: \${DB_DATABASE}

MYSQL_USER: \${DB_USERNAME}

MYSQL_PASSWORD: \${DB_PASSWORD}

MYSQL_ROOT_PASSWORD: \${DB_ROOT_PASSWORD}

volumes:

- mysql_data:/var/lib/mysql

healthcheck:

test:

- CMD
- mysqladmin
- ping
- -h
- localhost
- -u\${MYSQL_USER}
- -p\${MYSQL_PASSWORD}

redis:

image: redis:alpine

restart: unless-stopped

volumes:

- redis_data:/data

healthcheck:

test:

- CMD
- redis-cli
- ping

volumes:

app_public:

driver: local

app_storage:

driver: local

```
mysql_data:
  driver: local
redis_data:
  driver: local
networks: {}
```

.env

```
# IN application vars
TAG=latest
APP_URL=https://invoice.rfeyn.org
APP_KEY=base64:gDCwop+h+/KcHsrCzc7tt4lbqcqH8hzMnm3x3+8oGnM=
API_SECRET=Kikkers49$91
APP_ENV=production
APP_DEBUG=true
REQUIRE_HTTPS=false
PHANTOMJS_PDF_GENERATION=false #Turn this function off to generate pdf with snappdf
PDF_GENERATOR=snappdf
TRUSTED_PROXIES='*'

CACHE_DRIVER=redis
QUEUE_CONNECTION=redis
SESSION_DRIVER=redis

REDIS_HOST=redis
REDIS_PASSWORD=null
REDIS_PORT=6379

FILESYSTEM_DISK=debian_docker

# DB connection
DB_HOST=mysql
DB_PORT=3306
```

```
DB_DATABASE=ninjadplancer
DB_USERNAME=ninja
DB_PASSWORD=sf9823fsjlfkjwioefjlskfh0923jfklsdjf
DB_ROOT_PASSWORD=sdpoifj293fj09030rjnsnedfo3498t
DB_CONNECTION=mysql

# Create initial user
# Default to these values if empty
IN_USER_EMAIL=dp@p2pnode.org
IN_PASSWORD=nvQi5W14bl0npv
# IN_USER_EMAIL=
# IN_PASSWORD=

# Mail options
MAIL_MAILER=log
MAIL_HOST=smtp.porkbun.com
MAIL_PORT=587
MAIL_USERNAME=dp@p2pnode.org
MAIL_PASSWORD=PuV9#@pUca49C70e2V46&keg
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS='dp@p2pnode.org'
MAIL_FROM_NAME="DPLancer Invoices"

# MySQL
MYSQL_ROOT_PASSWORD=sdpoifj293fj09030rjnsnedfo3498t
MYSQL_USER=ninja
MYSQL_PASSWORD=sf9823fsjlfkjwioefjlskfh0923jfklsdjf
MYSQL_DATABASE=ninjadplancer

# GoCardless/Nordigen API key for banking integration
NORDIGEN_SECRET_ID=
NORDIGEN_SECRET_KEY=

IS_DOCKER=true
SCOUT_DRIVER=null
#SNAPPDF_CHROMIUM_PATH=/usr/bin/google-chrome-stable
```

2.

02. Payment Integrations (Mollie)

After you have successfully installed Invoice Ninja and created an account it is time to connect a payment integration with it. You don't need to have a payment integration to use Invoice Ninja, however if you went through the hassle of installing it you better do it ;) Yes the integrations are not fully open source but it's the best we have so far

1. Go to Settings --> Payment Settings --> Add Payment Gateway
2. Then choose a payment gateway that suits your needs. This could be for instance Stripe for USA and Mollie for European customers. I am from the Netherlands where most people use iDeal/Wero for their payments. Hence, I went with Mollie for my personal invoice ninja instance for my hospitality business. Hence, the following is a Mollie integration but it would work the same for any other payment gateway.
3. In the case of Mollie create an account and make sure you provide them with a business website or a proper description of your business. This is necessary for them to accept iDeal payments for instance.
4. If the iDeal payments are not accepted automatically after one day then I would recommend to contact support. This is what I did and they accepted it straight away.
5. If that is all set you need to obtain an API key and a Profile ID.
6. Inside of Mollie click on "Developers" that sits in the left bottom corner of your dashboard.
7. Then choose API Access Tokens -> + Create new Token
- 8.

03. Localization

This is a short page, but it is of importance! If you operate in country X you would like that the emails and client portals your clients interact with is in language X. This is where localization comes in.

1. Go to Settings -> Localization
2. Then select the right language, timezone and date format for the country you are operating in.