



# NodePoint IT System - Linux version 1.7.3

Patrick Lambert

<http://nodepoint.ca>

June 9, 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Requirements	6
1.2	Installation	6
1.3	Upgrading	6
1.4	Uninstalling	6
1.5	Troubleshooting	6
1.5.1	If you get a 500 server error	6
1.5.2	You see a list of files in your browser instead of the proper interface	7
1.5.3	If you get the error Could not access configuration file or initial settings don't seem to save	7
1.5.4	If you get the error Could not access database	7
1.5.5	If you forgot your admin password	7
1.5.6	If automation tasks do not run	7
1.5.7	If email notifications don't work	7
1.6	Best practices	7
1.6.1	Backups	7
1.6.2	Restores	8
1.6.3	Data integrity	8
1.7	Security concerns	8
1.8	Support	9
1.9	Additional customization	9
<b>2</b>	<b>Configuration</b>	<b>10</b>
2.1	Initial configuration	10
2.2	Configuration options	10
<b>3</b>	<b>Users and activity</b>	<b>14</b>
3.1	Users management	14
3.2	Active Directory integration	15
3.3	Reports	15

<b>4 Components</b>	<b>16</b>
4.1 Tickets Management	16
4.1.1 Custom forms	16
4.1.2 Favorites and assignment	17
4.1.3 Ticket routing	17
4.2 Support Articles	18
4.3 Time Tracking	18
4.4 Shoutbox	18
4.5 Tasks Management	18
4.6 Clients Directory	19
4.7 Secrets Vault	19
4.8 Inventory Control	20
4.9 Billing	21
4.10 Files Management	21
4.11 Automation	22
4.11.1 Backup	22
4.11.2 Email to Ticket	22
4.11.3 Bulk export	22
4.11.4 Users sync	22
4.11.5 Computers sync	22
4.11.6 Ticket expiration	23
4.11.7 Reminder notifications	23
4.11.8 ServiceNow CMDB	23
4.11.9 Update MOTD	23
4.11.10File expiration	23
4.11.11Log export	23
4.11.12CSV inventory	23
4.11.13CSV projects	23
4.11.14ODBC inventory	23
4.12 Checklists	24

<b>5</b>	<b>JSON API</b>	<b>25</b>
5.1	List users	25
5.2	List projects	25
5.3	Add new user	25
5.4	Verify user password	25
5.5	Change user password	25
5.6	List tickets by product	26
5.7	Show ticket	26
5.8	Add release	26
5.9	Add ticket	26
5.10	Update ticket	26
5.11	Assign ticket	27
5.12	Add comment	27
5.13	Show time tracking	27
5.14	Show billing information	27
5.15	Add task	28
5.16	List tasks	28
5.17	Add client	28
5.18	List clients	28
5.19	List client events	28
5.20	Add client event	29
5.21	Add item	29
5.22	List files	29
5.23	List items	29
5.24	Assign item	29
5.25	Return item	30
5.26	Update item	30
5.27	List checklists	30
5.28	Show checklist	30
5.29	Update checklist	30

5.30 Code samples . . . . .	31
5.30.1 List users . . . . .	31
5.30.2 Add ticket . . . . .	31
<b>6 License</b>	<b>32</b>

# 1 Introduction

NodePoint is an IT platform based on the Bootstrap framework providing a number of components for tickets management, inventory control, support articles, tasks and more. It is meant to be simple to setup and use, yet still offers most features one would expect in such a system. It is provided as freely available open source software under the MIT License.

Here are some of the available features:

- **Users management:** Allow users to register accounts, reset their passwords, email addresses and more.
- **Access level permissions:** Six access levels allow you to control who has access to add tickets, modify their statuses, manage projects, and so on.
- **Active Directory integration:** Use the stand alone users component or synchronize with your AD domain.
- **Projects and milestone control:** Add projects and create milestones against those projects.
- **Statistics and reports:** Monitor your instance through a number of reports.
- **Custom home page:** See the tickets you created, are assigned to, or follow on your home page.
- **Custom forms:** Use the default form or create custom ones to customize the ticket entry page.
- **Full text search and filters:** Filter tickets by product, status or do a custom search.
- **Follow and auto assignment:** Users can follow tickets, assign themselves to work on issues, or auto assign themselves to projects.
- **Comments and file uploads:** Leave comments or attach files related to tickets.
- **Knowledge base articles:** Create articles to build a knowledge base linked to your products.
- **Tasks management:** Create tasks and assign them to users, with due dates and completion rates.
- **Inventory control:** Add items that users can checkout, with full approval control.
- **Clients directory:** Make a clients listing which can then be used with tickets and items.
- **Time tracking:** Users assigned to tickets can enter the time they took to solve issues and track their time.
- **Easy to install:** NodePoint works on Windows and Linux, is easy to install, and can be configured in minutes.
- **Trivial to update:** See when an update is available and update your instance with a single copy operation.
- **Responsive design:** Using the Bootstrap framework, NodePoint works just as well on the desktop as on mobile.
- **Simple and elegant interface:** The interface is easy to understand and elegant for your users to use.
- **Email notifications:** NodePoint comes with built-in email notifications.
- **Automation engine:** Run background processes without user input.
- **JSON API:** Create custom scripts or link other systems to enter tickets automatically.

## 1.1 Requirements

NodePoint was created for small to medium sized organizations (less than 1,000 users, less than 10,000 tickets). It runs on a single server and does not require any external resource. Configuration settings are stored in the `../nodepoint.cfg` file while data is stored in a file database from a customizable location (the `../db` folder by default). As such, only two requirements are needed to install:

- A 64 bits Linux host.
- Apache web server.

## 1.2 Installation

Follow these steps to complete a new installation:

1. Download NodePoint from <http://nodepoint.ca> and untar the folder: `tar xvf nodepoint.tar`
2. Run `./setup.sh` as root, from the folder you just extracted. This will copy the files to the proper location, add some Apache configuration options, and setup the Automation schedule. Optionally, you may need to edit your Apache configuration or setup a virtual host.
3. Access the site from *localhost* (e.g. `http://localhost/nodepoint`) and complete the first time configuration (see below).

## 1.3 Upgrading

NodePoint is designed to be fully upgradable from any minor version to another. For example, `1.x.x` to `1.x.x`. To upgrade from a previous NodePoint version, simply copy the new `www` folder onto the old one. This will overwrite all product files but leave your configuration, data and uploads.

## 1.4 Uninstalling

You can remove a NodePoint installation from a host in two steps:

1. Remove the folder where your NodePoint files live (backup your database if you wish to retain your data).
2. Remove the virtual host entry from your web server.

## 1.5 Troubleshooting

### 1.5.1 If you get a 500 server error

Check in your server log what the likely error might be. Try to run `~/nodepoint/www/nodepoint.cgi` from a shell. Make sure you have a 64 bits OS.

### 1.5.2 You see a list of files in your browser instead of the proper interface

Make sure the `.htaccess` file is present, and that your Apache configuration allows overrides. Look for the line `AllowOverride All` in your default site configuration file.

### 1.5.3 If you get the error Could not access configuration file or initial settings don't seem to save

Make sure NodePoint has write access to `../nodepoint.cfg`. Make sure the file only contains one line with the text *dummy value*.

### 1.5.4 If you get the error Could not access database

Make sure NodePoint has write access to the database, by default `../db/nodepoint.db`.

### 1.5.5 If you forgot your admin password

Simply delete the `nodepoint.cfg` configuration file on the server and recreate it through the web interface from *localhost*.

### 1.5.6 If automation tasks do not run

Make sure the **nodepoint-automate** process is running every 5 minutes. You should have a listing under `crontab -L` that shows the proper path, under a user that has access to the NodePoint database file.

### 1.5.7 If email notifications don't work

This is typically because of permission issues. Make sure that:

- The user that NodePoint runs under has permission to make network connections.
- The firewall is allowing outgoing connections.
- The SMTP server is accepting connections from the NodePoint server and email address you set.

## 1.6 Best practices

### 1.6.1 Backups

NodePoint is likely to hold data you may not want to lose. The product stores information in three different locations:

First, configuration data is stored in the `nodepoint.cfg` file. There is very little information stored there, and you can view all of it on the **Settings** screen. Some of the information you may wish to backup includes:

- The administrator password.



- The API read, write and encryption keys.
- Information for your SMTP and Active Directory servers.
- Any other changes made such as default access levels, database location etc.

The most important data is held in the database file. This includes tickets, articles, user information, and so on. This file should be backed up on a regular basis. You can simply copy the file somewhere through a script, or as part of a backup strategy.

Finally, all uploads are stored in the `../uploads` folder. This includes both images used with products, along with files uploaded by users as part of ticket comments. These should also be backed up as part of any backup strategy you may have.

If you use the **Automation** component, a backup module is provided that will make regular backups of your database and uploads.

### 1.6.2 Restores

Restoring a NodePoint installation is fairly simple. Once you have finished a new installation, simply restore your backed up data before accessing the NodePoint site. There is nothing to restart and no service to worry about.

### 1.6.3 Data integrity

Data stored in the `nodepoint.db` file is stored in SQL format, using very efficient and proven technology. Locks are used to avoid data duplication. Data corruption is unlikely, but backups should still be done in the rare case that something happens to the file. Partial recovery is possible but not something to be attempted on a live system.

This does mean that you can access the data directly using a SQLite client. This however is highly discouraged without first contacting us for support. Doing the *vacuum* operation for example will corrupt your data.

## 1.7 Security concerns

By default, data stored in the database is accessible from anyone with login access to the server. As such, only administrators should be given physical access to the server used. User passwords are stored in a hashed format, both for the administrator and regular users. Administrative passwords for SMTP and automation tasks are stored in RC4 encrypted using the encryption key.

Using SSL is highly recommended and does not impact the operation of NodePoint at all. Simply add an encryption certificate to your web server and access the product normally.

The API provided uses a read and write key. This means that all operations are done as a special *api* user. Different amounts of secrecy can be used for the two keys, such as users having access to a script that exposes the read key, but the write key only being used on a server side script. If **User Impersonation** is configured, the API calls can specify a specific user for a number of operations as well.

NodePoint provides a file upload system. This can be used to upload project images, file attachments to comments, and file upload fields on custom forms. Users may use this feature to upload malicious files. You

should use an anti-virus software to scan for potential threats. You can also restrict the access level required to upload files on the **Settings** page, what file extensions are allowed, or disable uploads completely.

Finally, while NodePoint offers Active Directory integration, note that this is done using LDAP Simple Bind, not Kerberos, in order to allow both the Windows and Linux version to support this feature. LDAP over SSL can be implemented to provide encryption.

## 1.8 Support

If you need any functionality that is not covered by the base product, check out the **Resources** section of the web site where various scripts can be found. You can use the plugin options along with the JSON API to extend the functionality of the product.

Here are some examples of scripts available:

- Batch import of tickets from a CSV file.
- Desktop ticket support client.
- Create a ticket from a PowerShell prompt.

## 1.9 Additional customization

While NodePoint was created to support many features and provide several customization options, there are cases where you may have additional needs. This is why the source is freely available on GitHub for you to modify.

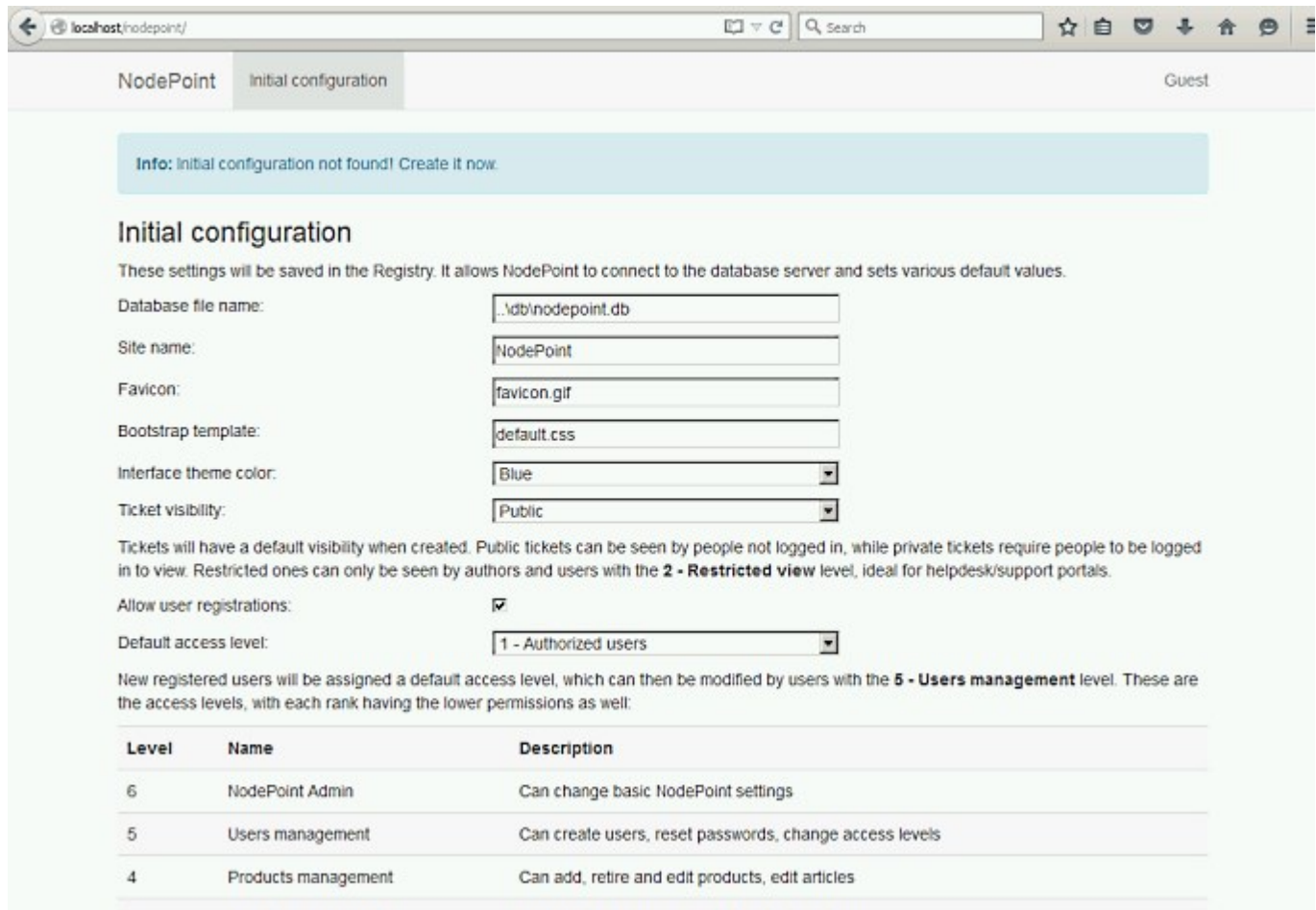
Alternatively, you can contact the author at `dendory@live.ca` if you have any suggestions for future versions of the product.

## 2 Configuration

Basic configuration options are defined the first time you access the product. The same information is available both from this initial configuration and at any point from the **Settings** page while logged in as the product administrator.

### 2.1 Initial configuration

When you first access NodePoint, you are presented with this screen:



Info: Initial configuration not found! Create it now.

### Initial configuration

These settings will be saved in the Registry. It allows NodePoint to connect to the database server and sets various default values.

Database file name:

Site name:

Favicon:

Bootstrap template:

Interface theme color:

Ticket visibility:

Tickets will have a default visibility when created. Public tickets can be seen by people not logged in, while private tickets require people to be logged in to view. Restricted ones can only be seen by authors and users with the **2 - Restricted view** level, ideal for helpdesk/support portals.

Allow user registrations: ☒

Default access level:

New registered users will be assigned a default access level, which can then be modified by users with the **5 - Users management** level. These are the access levels, with each rank having the lower permissions as well:

Level	Name	Description
6	NodePoint Admin	Can change basic NodePoint settings
5	Users management	Can create users, reset passwords, change access levels
4	Products management	Can add, retire and edit products, edit articles
3	Tickets management	Can create, update, delete tickets

The page includes reasonable defaults along with some explanations of what each option does. You should read them carefully. At a minimum, you must specify an administrator password. Do not lose it, as it is required to change the configuration at a later point. Not all configuration options are shown on that page.

### 2.2 Configuration options

These are the configuration options for NodePoint. They have to be defined during first use and can be modified by the product administrator later from the **Settings** tab.

- **Database file:** This is the NodePoint database file.
- **Upload folder:** Where product images and comment files are stored. Leave empty to disable uploads.
- **Admin name:** The product administrator.

- **Admin password:** The password for the administrator.
- **Site name:** The name shown at the top left of the interface.
- **Public notice:** This notice is shown on every page. Leave empty to remove the notice.
- **Email notifications signature:** Replace the default signature at the bottom of all notification emails with this text.
- **Bootstrap template:** This is a CSS file used by NodePoint. It should be a valid Bootstrap template, or could contain extra CSS elements you wish to use. The default file is `default.css` and contains a few such elements.
- **Interface theme color:** The color of panels in the interface.
- **Favicon:** This is the icon shown by web browsers in the URL bar.
- **Main page logo:** Image file shown on the login page.
- **Pinned article:** Article ID that will be pinned to the home page of all users when they log in.
- **Ticket visibility:** When set to *Public*, all tickets will be visible by logged in and guest users. When set to *Private*, people need to log in to view tickets. *Restricted* prevents people without at least the Restricted View access level from viewing others' tickets. This is a good choice for IT/Helpdesk systems.
- **Disable Getting Started screen:** Do not show the getting started information to users who haven't logged in before.
- **Hide closed tickets:** Do not show closed tickets on the Tickets page. This is mostly for performance and does not hide them from reports or prevents users from manually entering a closed ticket number.
- **Allow HTML in articles:** Do not escape quotes and tags from articles.
- **Allow registrations:** Allow guest users to register a new account. If set to no, the only way to add new accounts is if a user with Users Management access adds one.
- **Allow guest tickets:** Allow guest users to file tickets.
- **Maximum file upload:** Maximum file size for uploads to project images, ticket comments and the File Management component.
- **Must be assigned to project:** Project specific functions like adding releases, adding tasks, adding, deleting or viewing secrets, editing checklists or uploading files are only available to users auto-assigned to the project.
- **Allowed upload extensions:** A space separated list of file extensions allowed for uploads. All file types are allowed if left empty.
- **Number of row per page:** Number of rows to show per page on tables.
- **Session expiry time:** Number of hours before a user session is invalidated.
- **API read key:** Used for some API operations.
- **API write key:** Used for some API operations.
- **Encryption key:** Used to encrypt some configuration values along with secrets.
- **Allow user impersonation:** Allows API calls to specify a *from\_user* parameter when creating a ticket.
- **SMTP server:** The hostname for email notifications. Leave empty to disable email notifications.
- **SMTP port:** The port number for SMTP connections, defaults to 25.

- **SMTP username:** User for SMTP connections, if your mail server requires authentication.
- **SMTP password:** Password for SMTP connections, if your mail server requires authentication.
- **Support email:** The email from which email notifications come from.
- **Type of portal:** The type of items NodePoint should manage. This is purely a UI customization.
- **Active Directory server:** Enter your domain controller address to enable AD integration. Users will be created dynamically as they first log on, and passwords will be checked against AD using LDAP.
- **Active Directory domain:** The domain to use for AD integration (in NT4 format).
- **New users access level:** The access level for new users. The default is 1.
- **Can upload files:** The minimum access level a user must have to attach files to comments.
- **Can modify past ticket changes:** Below that level, users who edit tickets cannot change the previous description of the ticket.
- **Can edit custom forms / routing:** The minimum access level to create and edit custom ticket forms or ticket routing under the Tools menu.
- **Can view reports and statistics:** The minimum access level to access the statistics screen under the Tools menu.
- **Can assign tasks to users:** The minimum access level to assign tasks to users on the Projects page.
- **Can view user details:** Users of this access level or higher can view a list of users along with summary pages listing email addresses, access levels, tickets entered and checked out items.
- **Can view client details:** The minimum access level to view details about clients, including billing information.
- **Can add client events:** Users of this level can add and edit events.
- **Can configure automation:** Users of this level can configure automation modules.
- **Can add secrets:** Users of this level can add project secrets if the proper component is turned on.
- **Can view secrets:** Users of this level can view project secrets if the proper component is turned on.
- **Can create checklists:** Users of this level can create new checklists on a project's page.
- **Authentication plugin:** Use an external script or application to authenticate users. This can be used instead of the built-in users management or Active Directory integration if you have some other process to authenticate users. It must return 0 for success, 1 for failure. The following variables can be used: `%user%` and `%pass%`. For example: `python auth.py -userid %user% -passwd %pass%`
- **Notifications plugin:** Call an application or script when a notification is sent. You can use the following variables here: `%user%`, `%title%` and `%message%`. For example: `echo %message% >> ..\%user%.log`
- **Checkout plugin:** Call an application or script when an inventory item is checked out or approved. You can use the following variables here: `%user%` and `%serial%`. For example: `python add_license.py -item %serial% -user %user%`
- **Task completion plugin:** Call an application or script when a user completes (100%) a task assigned to them. Variables allowed are: `%user%`, `%task%`, `%product%` and `%due%`. For example: `echo %user% completed task %task% due on %due% >> tasks.log`
- **New tickets plugin:** You can call a script or application when a ticket is created. The available variables are `%ticket%`, `%user%`, `%title%`, `%description%`, `%product%` and `%release%`. For example: `curl --data 'ticketid=%ticket%&user=%user%' https://hr.company.com/track.cgi`

- **Ticket resolution plugin:** You can call a script or application every time a ticket is updated, along with the %user% who made the change, the new %status% of the %ticket%, the billable %client% and the %resolution%. Example: `echo %ticket%,%user%,%status%,%resolution%,%client% >> tickets.csv`
- **Components:** Turn component on or off. See chapter 4 for information about components.

### 3 Users and activity

NodePoint provides a simple way to manage users based on their access levels. Users have a specific level, which determines what shows up to them in the interface and what they can do. They can also enter an optional email address for notifications. Email addresses must be confirmed through the sending of an automated token when they first register, if email notifications are turned on. Users also have a password which can be changed by the user under the **Settings** page, through the *lost password* form, or reset by someone with the Users Management level. Users can be disabled but not deleted.

These are the access levels used by NodePoint along with a short description:

Level	Name	Description
6	NodePoint Admin	Can change basic NodePoint settings.
5	Users management	Can manage users, reset passwords, edit clients.
4	Projects management	Can add, retire and edit projects, edit articles and items.
3	Tickets management	Can create releases, update tickets, track time.
2	Restricted view	Can view statistics, restricted tickets and products.
1	Authorized users	Can create tickets and comments.
0	Unauthorized users	Can view private tickets.

NodePoint has a few special user accounts. The *admin* account is used as product administrator and acts differently than normal user accounts. It should only be used for administrative functions. *Guest* is used to specify people who are not logged in, and cannot be registered as a user account. The name *api* refers to any change made using the JSON API while *System* is used for automation. The *demo* account can be registered normally but cannot change its own password, and is meant for product demos.

#### 3.1 Users management

Management of user passwords and access levels can be done by anyone with access level 5 or higher. With the right access level, this panel will then be available in the **Tools** menu:

Users management			
Filter users by access level: All   0   1   2   3   4   5			
User name	Email	Level	Last login
arya	arya@seven-kingdoms.local	3	Thu May 7 06:21:52 2015
cersei	cersei@seven-kingdoms.local	3	Thu Nov 12

From there you can see a list of users, their email addresses, access levels, last logins along with the options to change their access levels and reset their passwords. Note that email notifications will only be sent out to users who confirmed their email addresses. This panel also allows passwords to be reset. You can see a summary page about a user by clicking on their names, and users can be disabled from that page.

Finally, new users can be created from that panel. If the **Allow Registrations** option is enabled, people



can also register new user accounts from the login screen.

Here are the character restrictions for users:

- **User names:** Between 3 and 50 characters, using alphanumeric characters, plus the dot, at, under-score and dash. User names are not case sensitive during login authentication.
- **Passwords:** At least 6 characters.
- **Email addresses:** At the most 99 characters.

## 3.2 Active Directory integration

Active Directory integration can be enabled in the initial configuration, which will link the users management panel to your domain controller. Passwords will no longer be stored in the NodePoint database. Once a user attempts to login, their credentials will be validated, then NodePoint will check if a user entry exists in the database. A new entry will be created on first login. Users still need to set their email addresses for email notifications to work. Access levels can still be assigned to users to control their access to the NodePoint interface.

Authentication uses the LDAP protocol. You must specify a valid domain controller hostname or IP address for logins to work. If a login fails, you can view the log on the **System log** page to see the error message.

## 3.3 Reports

User activity can be monitored through two graphs along with a number of reports. The first graph shows the number of tickets entered per day for the last 7 days. The second one shows an overview of all tickets based on status. A calendar will also show time spent on tickets, due tasks and item expiration dates. Reports can be shown on the screen or downloaded as CSV documents to open in a spreadsheet application. Both are available to users of level 2 and higher (configurable) on the **Statistics** page.





## 4 Components

NodePoint is divided into multiple components that you can enable or disable in the configuration as a product administrator. You can turn them on or off depending on your needs. For example, a help desk portal could have the **Tickets Management**, **Time Tracking** and **Support Articles** components enabled, while a project management portal could have only the **Tasks Management** component enabled.

### 4.1 Tickets Management

Users who have an access level of 1 or higher can enter tickets on the **Tickets** page. The default form asks for a title and description. Custom forms can also be designed and assigned to projects by users of access level 4 (configurable) and above. These custom forms can have up to 10 fields, each field can be a text area, a check box, satisfaction rating, priority, and so on. Projects that have a custom form assigned will display that form to users instead of the default one.

Users with access level 3 or above can modify tickets and be assigned to them. When a ticket is modified, the description is updated to show who modified it, and what parameters were changed, such as the title, description, status and resolution. Open tickets do not require a resolution, but each other status does. The list of tickets can be filtered based on project, status and so on.

#### 4.1.1 Custom forms

To use custom forms, follow these steps:

1. Add at least one project on the **Projects** page.
2. Create a custom form on the **Custom forms** page.
3. Assign that form to the project.

Here is an example of a custom form being built along with the result:

The image displays two side-by-side screenshots from the NodePoint IT System interface.

The left screenshot, titled "Custom form", shows the configuration page for a custom form. It includes fields for "Form name" (Helpdesk form 1) and "Product linked to this form" (Helpdesk for users). Below these is a table with columns "Field", "Question", and "Answer type". The table contains 9 rows of form fields:
 

Field	Question	Answer type
Title	The nature of your problem	Some text
1	Is this an urgent issue?	Yes / No
2	How many hours did you spend trying to solve it?	Number
3	Describe your issue:	Lots of text
4	Satisfaction weee	Satisfaction scale
5	Check?	Checkbox
6	True or false kinda thing..	True / False
7	Pick a date:	Date
8	Your IP:	IP address
9		Some text

 A "Save" button is at the bottom right.

The right screenshot, titled "Create a new ticket", shows the resulting form for the "Helpdesk for users" product. It includes a "Release" dropdown set to "1.0". The form fields correspond to the configuration:
 

- "The nature of your problem": Text input
- "Is this an urgent issue?": Radio buttons for "Yes" and "No"
- "How many hours did you spend trying to solve it?": Number input
- "Describe your issue:": Text area
- "Satisfaction weee": Satisfaction scale dropdown
- "Check?": Checkbox
- "True or false kinda thing..": Radio buttons for "True" and "False"
- "Pick a date:": Date input (format mm/dd/yyyy)
- "Your IP:": IP address input (value 10.10.1.1)

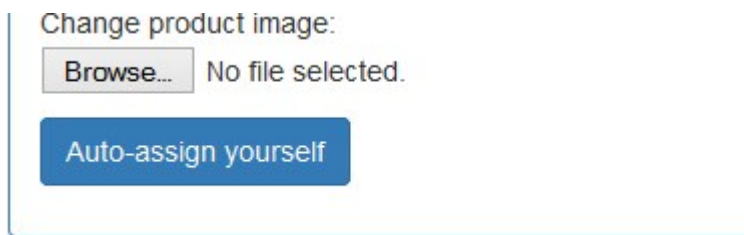
 A "Create ticket" button is at the bottom right.

### 4.1.2 Favorites and assignment

Any logged in user can favorite a ticket by clicking the proper option on the ticket's screen, which will add that ticket to their home screen for easy reference. This is meant as a way to follow an issue you are not actively working on.

Users with access level 3 or higher can assign themselves to a ticket by clicking the proper checkbox on the ticket's screen, or be assigned by a custom form. This is meant as a way to assign resources to a particular issue.

Auto assignment can also be used to assign specific users to any ticket filed against a project. You can add yourself to the auto-assign list of a product by clicking the button on the project's screen:



Both favorite and assigned tickets are listed on the user's home page. Finally, email notifications are sent out when a ticket is created, modified or a comment is added. The notifications are sent to the ticket author and users assigned to that ticket.

### 4.1.3 Ticket routing

A number of automatic actions can be applied to new tickets based on a number of conditions. Note that this is different than the scheduled background processes provided by the Automation component. Routing is done every time a new ticket is submitted through the web interface.

An unlimited number of routes can be created under the **Tools** menu by users with the proper access level. Each route has a priority and will be evaluated in that order. When you edit a route, you have to specify which conditions the ticket has to evaluate against, and which actions to apply if the conditions are true. Any condition left blank will not be evaluated.

Some conditions and actions accept the following variables:

- **%user%** - The user who created the ticket.
- **%priority%** - The priority of the ticket.
- **%ticket%** - The ticket ID number.
- **%assigned%** - The people auto-assigned to the ticket.
- **%title%** - The title of the ticket.
- **%description%** - The description of the ticket.
- **%product%** - The project ID the ticket is filed with.
- **%release%** - The release/milestone the ticket is associated to.
- **%field1%** - The content of field 1 (custom forms only).
- **%field2%** - The content of field 2 (custom forms only).

- **%field3%** - The content of field 3 (custom forms only).
- **%field4%** - The content of field 4 (custom forms only).
- **%field5%** - The content of field 5 (custom forms only).
- **%field6%** - The content of field 6 (custom forms only).
- **%field7%** - The content of field 7 (custom forms only).
- **%field8%** - The content of field 8 (custom forms only).
- **%field9%** - The content of field 9 (custom forms only).

## 4.2 Support Articles

The **Articles** page hosts articles that can be used to document common use cases, common issues, or to build a knowledge base for your organization. Articles are created in *draft* mode, which hides them from users with an access level lower than 4. Only once they are published are they visible to all users.

Users with at least access level 3 can link articles to tickets. These can be used to assign related documentation to issues, or used as categories. Linked tickets (except closed ones) are listed on the article's page, which makes it easy to track which tickets are assigned to a specific article.

## 4.3 Time Tracking

When updating a ticket, users can track the time they spent on the issue, which will be added to the time table for that ticket. A total of all the time spent on tickets by a specific user is shown on the **Settings** page. A report is also available with the total time spent by all users.

Time entries cannot be removed since it acts as a timestamp for work done. Corrections can be made by entering negative values.

## 4.4 Shoutbox

The shoutbox is a simple chat area on the home page that all users can use. The box shows the last few entries made by users, and individual entries can be deleted by users with access level 5 or higher.

## 4.5 Tasks Management

The tasks management component can be used to manage products or projects, create new tasks needed to be completed, and assign these tasks to users with due dates. Users with access level 4 (configurable) or higher can create new tasks on the **Projects** page.

If *Must be assigned to project* is on, users also need to be auto-assigned to that particular project to add tasks. New tasks created will send an email notification to the user assigned to the task, and all tasks will show on the user's home page. There, users can modify the completion rate for each item. Completed and overdue tasks will also be highlighted.

Tasks

Add a new task

Description

Assign user:

Uppercase

▼

Due by:

mm/dd/yyyy

Add task

Current tasks

Copy

CSV

PDF

Print

Search:

Task	Assigned to	Completion	Due by	
Check with dev is we can complete milestones in time	jaime	0%	Overdue	X
Send the RFP to contacts	tyrion	100%	Completed	X
Report for phase 1	cersei	0%	12/23/2015	X
Report for phase 2	cersei	30%	02/24/2016	X

Showing 1 to 4 of 4 entries

Previous

1

Next

## 4.6 Clients Directory

The clients directory can be used to keep track of clients or contacts from the **Clients** page. New clients can be added along with specific information such as email, phone number or notes by users of access level 5, details seen by users of level 2 (configurable) and higher. The following statuses can be assigned to clients:

- **Prospect:** A prospective client.
- **Supplier:** A vendor.
- **Contact:** A contact entry.
- **Paid:** A paid client.
- **Unpaid:** An unpaid or default client.
- **Closed:** A closed entry (no ticket can be filed against closed clients).

If the clients directory is enabled, the *Clients* field type can be added to custom ticket forms, which will show a drop down list of available clients when creating new tickets, to assign issues to specific clients. Events can be added by users of the right access level to log phone calls, emails and other communications. This can be used as a CRM portal to keep track of interactions with clients.

## 4.7 Secrets Vault

The Secrets Vault is a component that allows users assigned to a project to store secrets (credentials, phone numbers, anything that should be restricted) on a project basis. This will add a tab to any project

page where users of a specific level can add secrets, and others can view them. Accessing secrets is logged in the system log for auditing purposes.

Both the level to add and to view secrets are configured in the product base configuration. If *Must be assigned to project* is on, users also need to be auto-assigned to that particular project to be able to add or view secrets. The secrets are encrypted inside of the database using the encryption key. Changing that key will make the data irretrievable.

## 4.8 Inventory Control

The Inventory Control component allows users with access level 4 or higher to add items to the **Items** page. These items can be linked to projects or clients, and can be checked out by other users. Approval can also be required, in which case users will need to request access, and someone with access level 4 or higher will need to approve the request. This component can be used to keep track of physical items such as computers, keyboards and phones, or even software licenses.

Inventory items

Items awaiting approval

Copy CSV PDF Print

Search:

Type	Name	Serial	User	Action
Phone	iPhone 6	523-23563443-232	cersei	<div>Approve</div> <div>Deny</div>

Showing 1 to 1 of 1 entries

Previous 1 Next

Items list

Copy CSV PDF Print

Search:

Type	Name	Serial	Status
Blah wee blah	Logitech keyboard	000-111-222	Expired
Desktop	HP Pavilion dv6	824-dv6-0001	Checked out by: cersei
Desktop	HP Pavilion dv6	824-dv6-0002	Checked out by: jaime
Desktop	fdgd	fdgd	Expired
Desktop	fghg	45456434	Available
Mouse	Logitech mousepad	2463-51345-0001	Checked out by: tyron
Phone	Samsung S3	325-5427146-7527	Checked out by: rob
Phone	iPhone 6	523-23563443-232	Waiting approval for: cersei

Showing 1 to 8 of 8 entries

Previous 1 Next

When a user checks out an item, additional information such as item location, lock codes or other secret data can be displayed on the item's page and sent through email to that user, so that the person can retrieve it. This information field is otherwise hidden from other users. If approval is required, a notification will be sent to all available users with level 4 or higher. They can then decide to approve or deny the request. Items can have an expiration date, after which checkouts will be disabled.

Users with checked out items will see a list of items on their home page, and can then return the item, making it available to other users. A checkout history is kept for each item and shown on the item page, showing all past requests and events. Items can also be manually assigned to users, or made unavailable, preventing any user from requesting or checking them out.

An *items* field is available to be used in custom ticket forms, which will display a list of checked out items that the user can file tickets about. Finally, a checkout plugin can be used to accomplish some automated task when an item is checked out, approved or assigned to a user.

## 4.9 Billing

The Billing component allows you to assign tickets to clients with either a fixed rate or an hourly rate. As such, this component requires the Tickets Management and Clients Directory components to be enabled. For hourly rates, the Time Tracking component must also be enabled.

When editing a ticket, a user of level 3 or higher can select a client from the drop down list, among clients that have been entered in the directory. Once tickets have been assigned to a client, the list of tickets with hours spent are listed on that client's page. The cost and currency can also be adjusted on that page.

## 4.10 Files Management

NodePoint provides a files upload feature for things like project images and file attachments to comments. If you enable the Files Management component, an option will be available under the **Tools** menu for users to upload arbitrary files as well. These files will be listed on that page, along with a direct link to download them, which can be used to send to clients or partners who may not have a user account configured.

Available files					
Copy CSV PDF Print		Search: <input type="text"/>			
File name	File size	Uploaded by	Date	Direct link	
aws.png	4164	tyrion	Wed Dec 9 13:09:01 2015	<a href="#">6BA65D53-8073-1014-969B-FD23F6CC22CF</a>	X
compile.log	12292	admin	Wed Dec 9 13:37:44 2015	<a href="#">6E8A9C71-8077-1014-B5A8-8B8B4BAFF292</a>	X
Project report.pdf	524855	admin	Wed Dec 9 13:35:59 2015	<a href="#">2FEBE096-8077-1014-963F-D4FB42341703</a>	X
Showing 1 to 3 of 3 entries					Previous 1 Next

Note that the files uploaded here, along with those attached to projects or tickets, must be below the configured maximum file size value. Also, only users with an access level above the one configured in the initial settings will be able to upload files. The files themselves will be available to anyone with the link.

Files can also be uploaded on a project specific basis on the project page, and will show up both under *Tools* and that project's page. If *Must be assigned to project* is on, users also need to be auto-assigned to that particular project to upload files.

## 4.11 Automation

NodePoint ships with an automation engine called **nodepoint-automate**. This engine is set as a scheduled task during the installation process and controls the background automation modules, processes that can run without user intervention. Note that this is different from Ticket Routing which defines actions applied whenever someone submits a ticket. Automation modules instead run in the background on a specific schedule.

Each module can be enabled or disabled individually, and scheduled on its own schedule of 5 minutes, 15 minutes, hourly, daily or weekly. Configuration is done under the **Tools** menu by users with the proper access level. A log of these processes is also available there.

Note that some tasks may take a significant amount of time to run and slow the interface down for users while they are running. It is recommended that you set up resource intensive tasks outside of peak hours. You can also run the automation binary manually from the command line to get an idea of the time delay, or to troubleshoot any unexpected error. Clicking the *Process all modules on next run* button will run every enabled module regardless of whether their schedule is up or not.

These are the available automation modules:

### 4.11.1 Backup

This module allows you to backup the database along with the uploads folder to another location for safe keeping. The folder must be a local or network path, and the type of backup can be a single archive file or a series of time stamped files.

### 4.11.2 Email to Ticket

This module can fetch emails from an IMAP account and turn them into tickets automatically. The mail server and inbox information must be provided in order to log in, along with information about the ticket to create, and whether to delete the email afterward.

### 4.11.3 Bulk export

This module will export a specific table to a file location automatically in CSV format. Specify the full path of a local file and the table to export.

### 4.11.4 Users sync

This module can keep the internal users list in sync with your Active Directory server. This requires credentials of a user with object listing rights. The Base DN should be the OU where your users are kept, and the filter can be used to filter specific object types. Emails can optionally be updated for all users as well.

### 4.11.5 Computers sync

This module will list computer objects from your Active Directory server and create entries in the Inventory Control component. The Base DN should be the OU where your computers are kept. The serial will be set to the machine's hostname. This requires credentials of a user with object listing rights.



#### 4.11.6 Ticket expiration

This module interacts with active tickets that were modified more than x days ago. It can notify assigned users, and close the ticket.

#### 4.11.7 Reminder notifications

This module will notify users about expired items checked out, overdue tasks and tickets requiring attention.

#### 4.11.8 ServiceNow CMDB

This module can import CMDB data from a ServiceNow instance and import the data into the Inventory Control component. You need to enter valid credentials and the name of the table to read, along with the table header mappings.

#### 4.11.9 Update MOTD

This module will dynamically update the MOTD from a text file.

#### 4.11.10 File expiration

This module can remove files uploaded using the Files component after a set amount of days.

#### 4.11.11 Log export

This module will export the system log, access log and automation log to a CSV file, and optionally clear the logs afterward.

#### 4.11.12 CSV inventory

This module will import items from a CSV file. You must specify the file name and the column mappings.

#### 4.11.13 CSV projects

This module fetches projects, releases and auto-assignments from a CSV file. You must specify column mappings and whether current assignments should be overwritten. The CSV should contain one line per release. Assignments must be comma separated usernames. Non-existent projects or releases will be created.

#### 4.11.14 ODBC inventory

This module will connect to an ODBC database using the specified DSN or connection string and the credentials provided. You also need to supply the column numbers (in order retrieved by a SELECT statement, starting at 0) to map against the inventory component.



## 4.12 Checklists

Checklists are key/value data pairs that can be added to custom tables. These tables are linked to specific projects on that project's page. Each checklist can given a name with access level restrictions for who can read the values, modify the values, and edit the keys. This component is useful to create checklists related to your projects, or keep arbitrary data easily accessible.

If *Must be assigned to project* is on, users also need to be auto-assigned to that particular project to edit checklists.

## 5 JSON API

NodePoint provides a JSON based REST API that can be used to accomplish some basic tasks. This can be useful to integrate the system with existing products, or to use with custom scripts. Each API call requires either the *read* or *write* key. The calls can be made using GET or POST.

Here is a list of commands. Arguments in *italics* are optional:

### 5.1 List users

- api=list\_users
- key=<read key>

### 5.2 List projects

- api=list\_products
- key=<read key>

### 5.3 Add new user

- api=add\_user
- key=<write key>
- user=<user name>
- password=<password>
- email=<email>

### 5.4 Verify user password

- api=verify\_password
- key=<read key>
- user=<user name>
- password=<password>

### 5.5 Change user password

- api=change\_password
- key=<write key>
- user=<user name>
- password=<password>

## 5.6 List tickets by product

- api=list\_tickets
- key=<read key>
- product\_id=<product id>

## 5.7 Show ticket

- api=show\_ticket
- key=<read key>
- id=<ticket id>

## 5.8 Add release

- api=add\_release
- key=<write key>
- product\_id=<product id>
- release\_id=<release id>
- notes=<notes>

## 5.9 Add ticket

- api=add\_ticket
- key=<write key>
- product\_id=<product id>
- release\_id=<release id>
- title=<ticket title>
- description=<ticket description>
- *priority*=<Low|Normal|High>
- *from\_user*=<valid user>

## 5.10 Update ticket

- api=update\_ticket
- key=<write key>
- id=<ticket id>

- *summary*=<summary of changes>
- *status*=<New|Open|Invalid|Hold|Duplicate|Resolved|Closed>
- *resolution*=<new resolution>
- *priority*=<Low|Normal|High>
- *time\_spent*=<number of hours>
- *from\_user*=<valid user>

## 5.11 Assign ticket

- *api*=assign\_ticket
- *key*=<write key>
- *id*=<ticket id>
- *user*=<valid user>
- *from\_user*=<valid user>

## 5.12 Add comment

- *api*=add\_comment
- *key*=<write key>
- *id*=<ticket id>
- *comment*=<comment>

## 5.13 Show time tracking

- *api*=show\_time
- *id*=<ticket id>
- *key*=<read key>

## 5.14 Show billing information

- *api*=show\_billing
- *client*=<client name>
- *key*=<read key>

## 5.15 Add task

- api=add\_task
- key=<write key>
- product\_id=<product id>
- description=<task description>
- user=<assigned user>
- due=<mm/dd/yyyy>

## 5.16 List tasks

- api=list\_tasks
- key=<read key>
- user=<user>

## 5.17 Add client

- api=add\_client
- key=<write key>
- name=<client name>
- status=<client status>
- contact=<contact address>
- *notes*=<*notes*>

## 5.18 List clients

- api=list\_clients
- key=<read key>

## 5.19 List client events

- api=list\_events
- key=<read key>
- client\_id=<client id>

## 5.20 Add client event

- api=add\_event
- key=<write key>
- client\_id=<client id>
- summary=<event summary>
- type=<event type>
- notes=<notes>
- from\_user=<valid user>

## 5.21 Add item

- api=add\_item
- key=<write key>
- product\_id=<0|product id>
- client\_id=<0|client id>
- approval=<0|1>
- type=<item type>
- name=<item name>
- serial=<serial number>
- info=<additional information>

## 5.22 List files

- api=list\_files
- key=<read key>

## 5.23 List items

- api=list\_items
- key=<read key>

## 5.24 Assign item

- api=assign\_item
- key=<write key>
- id=<item id>
- user=<user>

## 5.25 Return item

- api=return\_item
- key=<write key>
- id=<item id>

## 5.26 Update item

- api=update\_item
- key=<write key>
- id=<item id>
- *product\_id*=<product id>
- *client\_id*=<client id>
- *approval*=<0|1>
- *type*=<item type>
- *name*=<item name>
- *serial*=<serial number>
- *info*=<additional information>
- *expiration*=<mm/dd/yyyy>

## 5.27 List checklists

- api=list\_checklists
- key=<read key>

## 5.28 Show checklist

- api=show\_checklist
- key=<read key>
- name=<checklist name>

## 5.29 Update checklist

- api=update\_checklist
- key=<read key>
- key\_id=<checklist key to update>
- value=<new value>

## 5.30 Code samples

The following code snippets are Python 3.x examples of how to access the API.

### 5.30.1 List users

This script connects to a local NodePoint instance and lists all the users:

```
import json
import urllib.request
import urllib.parse

post_params = {'api': 'list_users', 'key': '0qvG4DkOz8NZnkuhn4aczck7r3qQVYS4'}

post_args = urllib.parse.urlencode(post_params)
data = post_args.encode()
stream = urllib.request.urlopen('http://localhost/nodepoint/', data)
result = stream.read()
charset = stream.info().get_param('charset', 'utf8')
users = json.loads(result.decode(charset))
for user in users['users']:
    print(user['name'])
```

### 5.30.2 Add ticket

This script connects to a local NodePoint instance and adds a ticket:

```
import json
import urllib.request
import urllib.parse

post_params = {'api': "add_ticket", 'key': "c5BT44108cYxBRHOYLnykazACsjkkQP3",
               'product_id': "1", 'release_id': "1.0", 'title': "This is a test ticket",
               'description': "Some description"}

data = urllib.parse.urlencode(post_params)
conn = urllib.request.urlopen("http://127.0.0.1/nodepoint/?" + data)
stream = conn.read()
result = json.loads(stream.decode(conn.info().get_param('charset', 'utf8')))
print(str(result))
```



## 6 License

The MIT License (MIT)

Copyright © 2014-2016 Patrick Lambert

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.