



NodePoint Manual - Windows version 1.3.2

Patrick Lambert

<http://nodepoint.ca>

June 25, 2015

Contents

1	Introduction	3
1.1	Requirements	4
1.2	Installation	4
1.3	Upgrading	4
1.4	Troubleshooting	4
1.4.1	If setup.bat fails	4
1.4.2	If you get the error Could not access Registry	4
1.4.3	If you get the error Could not access database file	5
1.4.4	If the server tries to download the file instead of executing it	5
1.4.5	If you get the error The page you are requesting cannot be served because of the ISAPI and CGI Restriction list settings on the Web server	6
1.4.6	If you forgot your admin password	6
1.4.7	If email notifications don't work	6
1.5	Best practices	6
1.5.1	Backups	6
1.5.2	Restores	7
1.5.3	Data integrity	7
1.6	Security concerns	7
1.7	Support	7
2	Configuration	8
2.1	Initial configuration	8
2.2	Configuration options	8
3	Users and activity	10
3.1	Users management	10
3.2	Active Directory integration	11
3.3	Reports	11
4	Components	12

4.1	Tickets Management	12
4.1.1	Custom forms	12
4.1.2	Following and assignment	13
4.2	Support Articles	13
4.3	Time Tracking	13
4.4	Shoutbox	13
4.5	Tasks Management	14
4.6	Clients Directory	14
5	JSON API	15
5.1	Show ticket	15
5.2	List tickets by product	15
5.3	List users	15
5.4	Verify user password	15
5.5	Change user password	15
5.6	Add new user	16
5.7	Add ticket	16
5.8	Add comment	16
5.9	Add release	16
6	License	17

1 Introduction

NodePoint is a ticket management platform based on the Bootstrap framework. 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 products, and so on.
- **Active Directory integration:** Use the stand alone users module or synchronize with your AD domain.
- **Product and release control:** Add products and create releases against those products.
- **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.
- **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.
- **Full text search and filters:** Filter tickets by product, status or do a custom search.
- **Time tracking:** Users assigned to tickets can enter the time they took to solve issues and track their time.
- **Follow and auto assignment:** Users can follow tickets, assign themselves to work on issues, or auto assign themselves to products.
- **Comments and file uploads:** Leave comments or attach files related to tickets.
- **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 a built-in email notification module to link with your email system.
- **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 5,000 users, less than 100,000 tickets). It runs on a single server and does not require any external resource. Configuration settings are stored in the Windows Registry 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 Windows host (7, 8, 10, Server 2008, Server 2012 should all work).
- IIS installed with *CGI* and *ISAPI Extensions* features.

1.2 Installation

Follow these steps to complete a new installation:

1. Download NodePoint from <http://nodepoint.ca> and unzip the folder where you want files to live, for example `C:\nodepoint`.
2. Run `setup.bat` and follow the instructions. This will create a new user for NodePoint to run under, configure IIS, and restart the web server.
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 Troubleshooting

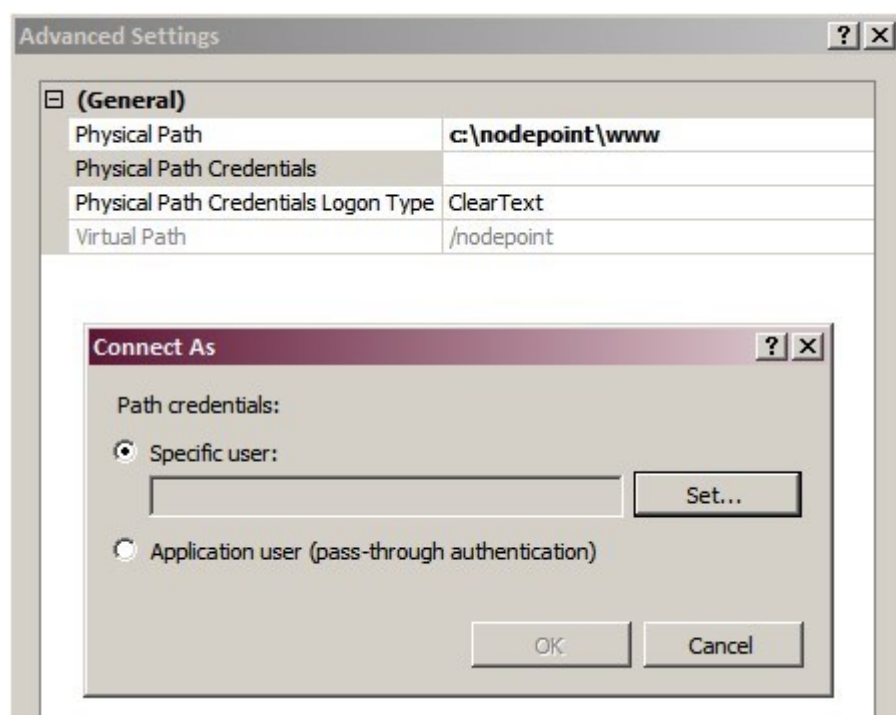
1.4.1 If setup.bat fails

Make sure you are running it as a local administrator. Make sure the file `%windir%\system32\inetsrv\appcmd.exe` exists, and that your web site is named *Default Web Site*. Also try running `setup` from the installation folder. Optionally, you may have to modify the script to suit your environment.

1.4.2 If you get the error Could not access Registry

Make sure the virtual folder has credentials that has access to the Registry. Go to *Administrative Tools -> IIS Manager*, then select the NodePoint site under *Default Web Site* and go to Advanced Settings. Once there, select a Specific User that has access to the Registry.

This can be your own user name, or you can create a dedicated user in *Administrative Tools -> Computer Management -> Local Users and Groups*. Make sure that user is part of the Administrators local group, and that you restart the IIS server after:

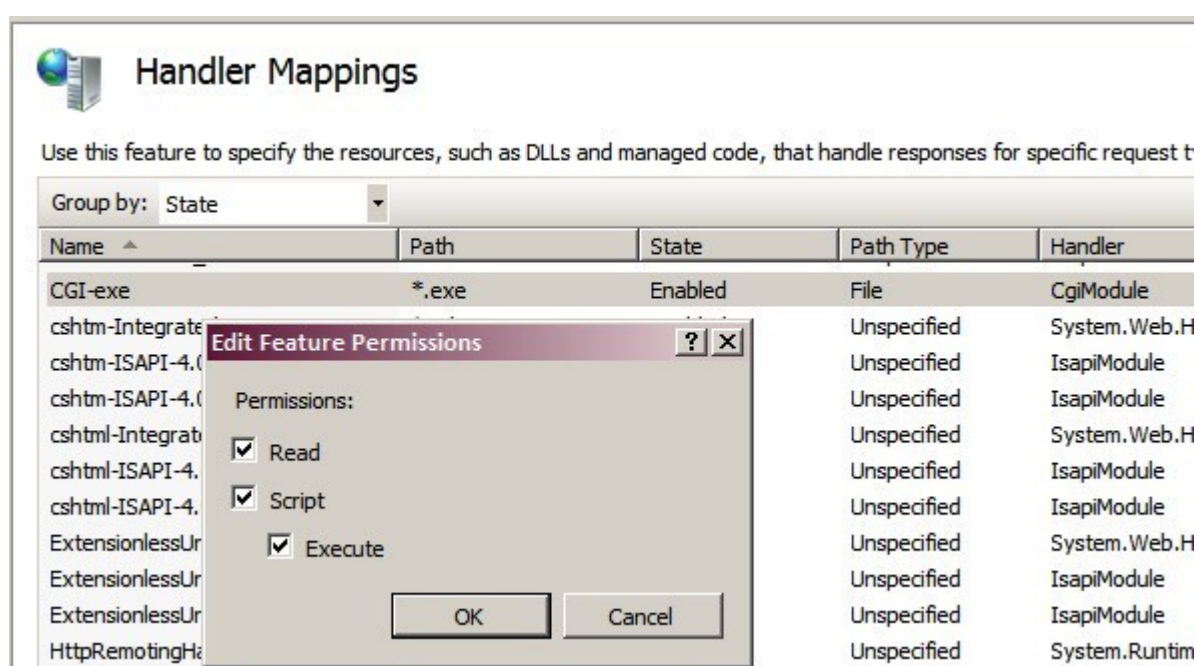


1.4.3 If you get the error Could not access database file

Make sure that NodePoint has Read/Write access to the database file specified in the initial configuration. By default this is C:\nodepoint\db\nodepoint.db. This should be the user you configured in the initial setup. You can use Windows Explorer to right click on the file and check under the Security tab.

1.4.4 If the server tries to download the file instead of executing it

Make sure you have the *CGI and ISAPI Extensions* IIS features installed and enable **Execute** under *IIS Manager -> Handler Mappings -> CGI-exe -> Edit Feature Permissions*:



1.4.5 If you get the error The page you are requesting cannot be served because of the ISAPI and CGI Restriction list settings on the Web server

Go to *IIS Manager -> ISAPI and CGI Restrictions* and make sure NodePoint is listed. If not, click on **Edit Feature Settings** and enable unspecified CGI modules.

1.4.6 If you forgot your admin password

You will need to remove the initial configuration settings and recreate them in order to recover access. This configuration is stored locally on the server inside the Windows Registry. Use `regedit.exe` and navigate to `HKLM/SOFTWARE/NodePoint`.

1.4.7 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.

1.5 Best practices

1.5.1 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 Windows Registry. 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 and write 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.

1.5.2 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.

1.5.3 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.

1.6 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.

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 as normally.

NodePoint is a stateless system, which means that users are re-authenticated at every operation. This means that things like cookie highjacking or manipulation will not allow a user to gain access if their password becomes invalid. While there is no ban or lock function in the product, a user with a proper access level can reset the password of another user which effectively locks them out. Note that users cannot be deleted in order to avoid many items and operations becoming orphaned.

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.

If you specify a SMTP server for email notifications along with a username and password, these will be stored in plain text. This is unavoidable for NodePoint to be able to log into your email server. It is recommended that you use IP based filtering instead of a specific account.

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.

1.7 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 also contact the support team at support@nodepoint.ca if you have any request or question.

2 Configuration

Basic configuration options are defined the first time you access the product. To prevent outsiders to reset your configuration in the event that the data is lost, the initial configuration must be done from the local host. 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

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.

2.2 Configuration options

These are the default configuration values 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.
- **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.
- **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.
- **Favicon:** This is the icon shown by web browsers in the URL bar.
- **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.
- **Default access level:** The access level for new users. The default is 1.
- **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.
- **API read key:** Used for some API operations.
- **API write key:** Used for some API operations.
- **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.
- **External 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`
- **Upload folder:** Where product images and comment files are stored. Leave empty to disable uploads.
- **Minimum upload level:** The minimum access level a user must have to attach files to comments.
- **Items managed:** The type of items NodePoint should manage. This is purely a UI customization.
- **Custom ticket field:** The name of the third ticket field in the default form (after *title* and *description*). This can be used to ask users who fill in tickets to list related tickets, or any other information relevant to your particular installation.
- **Custom ticket type:** The type of field. If *text*, then the entries will be shown as text. If *URL*, then links will be assumed. If *checkbox*, then users will have to select yes or no.
- **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).
- **Components:** Activate components that you require. This does not remove their functionality, only hiding them from the interface.

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, or reset by someone with the Users Management level.

These are the access levels used by NodePoint:

Level	Name	Description
6	NodePoint Admin	Can change basic NodePoint settings.
5	Users management	Can manage users, reset passwords, edit clients.
4	Products management	Can add, retire and edit products, edit articles.
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. 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 **Settings** page:

Users management					
Filter users by access level: All 0 1 2 3 4 5					
User name	Email	Level	Change access level	Reset password	Last login
test		1	Change access level	Reset password	Tue May 19 06:55:51 2015
tyrion	tyrion@seven-kingdoms.local	4	Change access level	Reset password	Fri May 15 11:57:04 2015
cersei	cersei@seven-kingdoms.local	3	Change access level	Reset password	Thu Apr 16 11:45:36 2015

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.

Note that while there is no lock or ban function, resetting a password accomplishes the same thing. A reset can also be done for a user who loses their password as well.

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

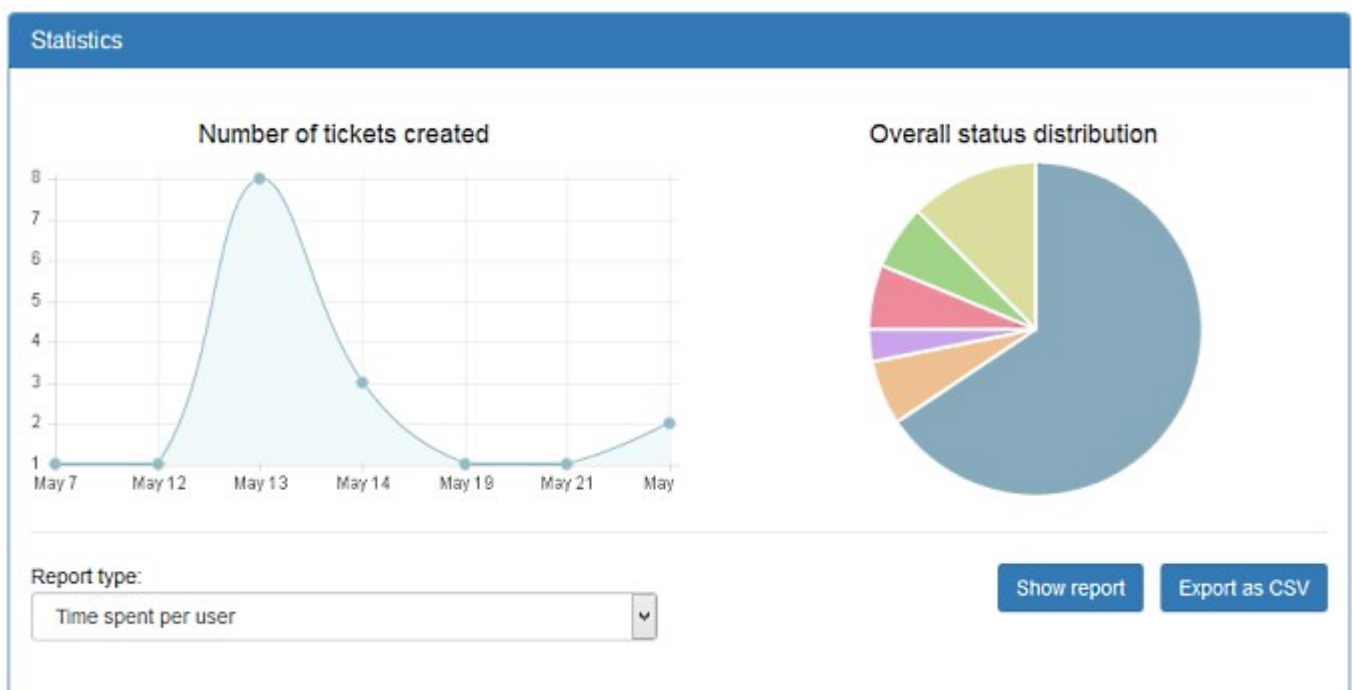
3.2 Active Directory integration

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 **Settings** 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. 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 on the **Settings** 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, description and a custom entry type, configurable in the initial settings. Custom forms can also be designed and assigned to products by users of access level 4 and above. These custom forms can have up to 10 fields, each field can be a text area, a check box, satisfaction rating, and so on. Products 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. 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 every other status does. The list of tickets can be filtered based on product, status and so on.

4.1.1 Custom forms

To use custom forms, follow these steps:

1. Add at least one product on the **Products** page.
2. Create a custom form on the **Settings** page.
3. Assign that form to the product.

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

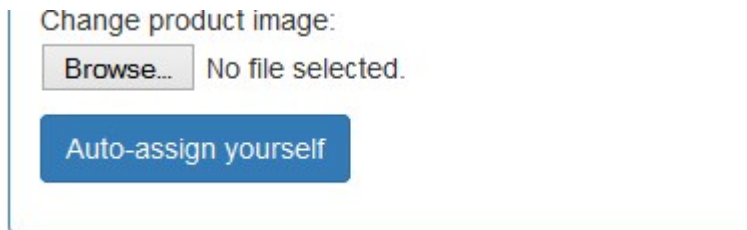
The image displays two side-by-side screenshots from the NodePoint application. The left screenshot, titled 'Custom form', shows the configuration interface for creating a custom form. It includes fields for 'Form name' (Helpdesk form 1) and 'Product linked to this form' (Helpdesk for users). Below these is a table with columns 'Field', 'Question', and 'Answer type'. The table contains 9 rows of fields: 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), and 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 contains the same 9 fields as the configuration: Title (The nature of your problem), 1 (Is this an urgent issue? with radio buttons for Yes/No), 2 (How many hours did you spend trying to solve it?), 3 (Describe your issue:), 4 (Satisfaction weee with a dropdown), 5 (Check? with a checkbox), 6 (True or false kinda thing. with radio buttons for True/False), 7 (Pick a date:), 8 (Your IP:), and 9 (Some text). A 'Create ticket' button is at the bottom right.

4.1.2 Following and assignment

Any logged in user can follow 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. 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 product. You can add yourself to the auto-assign list of a product by clicking the button on the product's screen:



Both followed 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.

4.2 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 2 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 or higher can create new tasks on the **Products** page:

Tasks

Add a new task

Assign user:

test

Due by:

mm/dd/yyyy

Add task

Current tasks

Task	Assigned to	Completion	Due by	
Setting up the project management calendar	tyrion	100%	Completed	Delete
Recruiting contractors for the work	cersei	0%	07/01/2015	Delete
Setting up the backend server for the work to begin	tyrion	100%	Completed	Delete
Implementation of the web scripts - phase 1	tywin	30%	08/12/2015	Delete
Implementation of the web scripts - phase 2	tywin	0%	10/17/2015	Delete

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.

4.6 Clients Directory

The clients directory can be used to keep track of clients or contacts from the **Settings** 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 and higher. The following statuses can be assigned to clients:

- **Prospect:** A prospective client.
- **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. This can be used as a CRM portal to keep track of interactions with clients.

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:

5.1 Show ticket

- api=show_ticket
- key=<read key>
- id=<ticket id>

5.2 List tickets by product

- api=list_tickets
- key=<read key>
- product_id=<product id>

5.3 List users

- api=list_users
- key=<read key>

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 Add new user

- api=add_user
- key=<write key>
- user=<user name>
- password=<password>
- email=<email>

5.7 Add ticket

- api=add_ticket
- key=<write key>
- product_id=<product id>
- release_id=<release id>
- title=<ticket title>
- description=<ticket description>
- custom=<custom field>
- from_user=<valid user>

5.8 Add comment

- api=add_comment
- key=<write key>
- id=<ticket id>
- comment=<comment>

5.9 Add release

- api=add_release
- key=<write key>
- product_id=<product id>
- release_id=<release id>
- notes=<notes>

6 License

The MIT License (MIT)

Copyright © 2014-2015 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.