

# HERALD DASHBOARD

## CONFIGURATION MANUAL V1.5

Herald by DEVOTED

[www.heraldservices.net](http://www.heraldservices.net)

[www.devoted.ba](http://www.devoted.ba)



## TABLE OF CONTENTS

1	INTRODUCTION.....	5
1.1	About the platform.....	5
1.1.1	General concept.....	5
1.1.2	Organizations.....	7
1.1.3	Groups .....	7
1.1.4	Users & Roles.....	7
1.1.5	Integrations .....	7
1.1.6	Triggers .....	8
1.1.7	Variables .....	8
1.1.8	Templates .....	8
1.1.9	Routing Rules.....	9
1.1.10	Instant Notifications .....	9
2	CONFIGURATION .....	10
2.1	Herald Dashboard web page.....	10
2.2	Register new account .....	10
2.3	Forgot password .....	10
2.4	Login .....	11
2.5	Set personal data .....	12
2.6	Organizations .....	13
2.6.1	About .....	13
2.6.2	Add new Organization.....	13
2.6.3	Pin Organization .....	14
2.6.4	Edit Organization .....	15
2.6.5	Delete Organization .....	15
2.7	Groups.....	16
2.7.1	About .....	16
2.7.2	Add new Group .....	16
2.7.3	Edit Group.....	17
2.7.4	Delete Group .....	18
2.8	Users.....	19
2.8.1	About .....	19
2.8.2	User Roles .....	19
2.8.3	Invite / Add Users .....	20
2.8.4	Edit User .....	22
2.8.5	Delete User .....	22
2.9	Integrations .....	24



2.9.1	About .....	24
2.9.2	Integrations overview .....	24
2.9.3	Add S7-1x00 (Direct) Integration.....	25
2.9.4	Add API (Server) Integration .....	33
2.9.5	Ping Integration .....	37
2.9.6	Activate/Deactivate Integration .....	37
2.9.7	Download Integration .....	38
2.9.8	Edit Integration.....	38
2.9.9	Delete Integration .....	39
2.10	Time-triggers.....	40
2.10.1	About.....	40
2.10.2	Add new Time-trigger.....	41
2.10.3	Edit Time-trigger.....	42
2.10.4	Delete Time-trigger .....	42
2.11	Triggers.....	43
2.11.1	Add new Trigger.....	43
2.11.2	Edit Trigger .....	46
2.11.3	Delete Trigger .....	46
2.12	Variables.....	47
2.12.1	Add new Variable.....	47
2.12.2	Edit Variable .....	50
2.12.3	Delete Variable .....	50
2.13	Templates.....	51
2.13.1	About .....	51
2.13.2	Templates overview .....	51
2.13.3	Add new Template .....	52
2.13.4	Edit Template.....	53
2.13.5	Delete Template .....	54
2.14	Routing Rules .....	55
2.14.1	About.....	55
2.14.2	Routing Rules overview .....	55
2.14.3	Add new Routing Rule .....	56
2.14.4	Edit Routing Rule .....	57
2.14.5	Delete Routing Rule.....	57
2.15	Instant Notifications .....	58
2.15.1	About.....	58
2.15.2	Send new Instant Notification .....	58
3	SUBSCRIPTIONS.....	59



3.1	About .....	59
3.2	Subscriptions overview .....	60
3.3	Balance overview .....	61
3.3.1	Add to Organization's balance.....	62
4	CONCLUSION.....	64



## 1 INTRODUCTION

DEVOTED Ltd. developed a notification platform named **Herald** (will be referred to as ‘the platform’ in this document) as a commercial product which includes two major modules:

1. [Herald App](#) – android application used for utilizing the platform’s app notification service,
2. [Herald Dashboard](#) – commercial online web application used for configuration of the platform.

Besides being an instruction manual on how to configure the platform using Herald Dashboard, this document gives an in detail explanation of the platform’s capabilities as well as the corresponding pricing.

### 1.1 About the platform

The platform supports integration of PLCs, IoTs, SCADAs and software applications, in order to receive notifications on events in corresponding plants or businesses. Android App, Email and SMS notifications are supported.

[Herald Dashboard](#) is a web application which serves as the platform’s central data-management entity. Among other, it is used to define events which users should get notified about, configure notifications’ content and how and when notifications are routed to users.

[Herald App](#) is an android application (can be downloaded for free from Google Play) which allows users to receive notifications as configured through Herald Dashboard. Inside the App, notifications can be searched, filtered, acknowledged and archived. Error, warning and info notifications are available.

#### 1.1.1 General concept

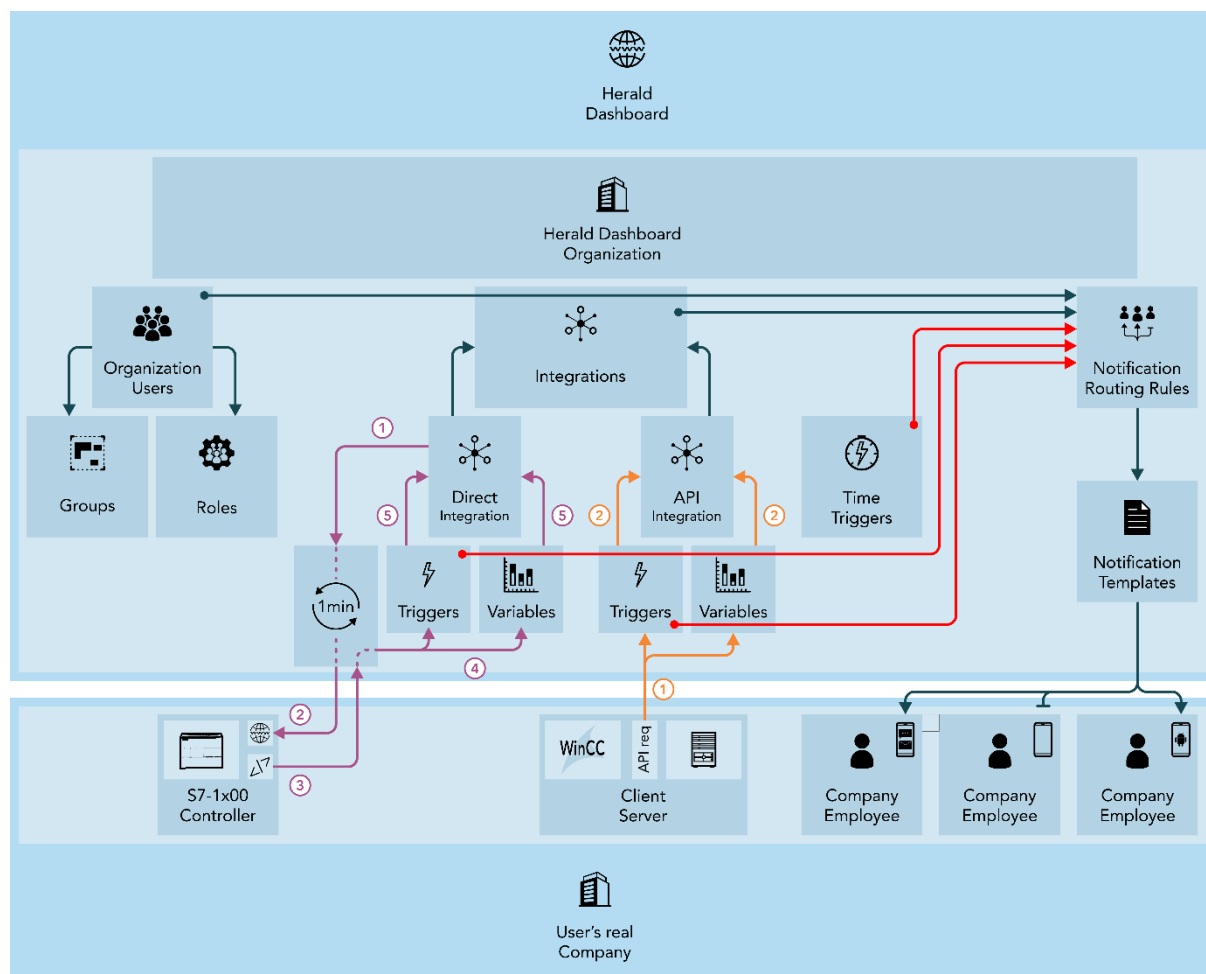


Figure 1.1 – Schematic illustration of the general concept of the platform



Once logged into Herald Dashboard, the user can start configuring the platform. Figure 1.1 shows a schematic illustration of the general concept of how the platform functions. Here, only the basics will be laid out. Details on the configuration process will be explained in sections to come.

Within the Herald Dashboard environment, a user can define an **Organization** (corresponds to a real-life company), invite **Users** (company employees) and assign them to different **Groups** and **Roles**. Since this is a notification platform, the user must define sources of events which he/she would like to notify employees about. These sources are called **Integrations**. To define the notifications' content and type (info, warning or error), the user must add **Notification Templates**. At last, to define when notifications are sent and to whom, the user must create **Notification Routing Rules**. So, in very few words, detected events from defined Integrations will trigger sending of notifications with content as defined in Templates, to Users as defined in Routing Rules.

An overview schematic of a real-life scenario is shown in Figure 1.1. On the bottom, a real life company is presented in form of a blue tile. Within the company, there is a PLC controller which is, let's say, responsible for controlling a production process. Also, there is a server which might be used for hosting an intranet website or a public web-application or similar. Naturally, there are the company employees. Basically, the objective here is to notify the company employees on events which are happening within the PLC and/or Server. These events could be represented in form of error, warning or info messages, indicating the state of the managed processes.

To achieve this goal, the PLC and Server must be integrated with the platform. This integration is done through Herald Dashboard. Within the Dashboard environment, an **Organization** is the virtual representation of a real-life company. The Organization is displayed in form of a blue tile on the top of Figure 1.1. For every real-life user that one would like to notify using the platform, organization **Users** must be added. They can be assigned to **Groups** and **Roles**. Groups can be created to the preferences and needs of the user/company. Users can be assigned to multiple Groups. By default, three Roles are available: Administrator, Operator and Regular user. Every employee which is added as a user is added with one of these three available Roles where each has a different set of permissions.

Devices such as PLCs and Servers are represented with **Integrations**. A PLC device is represented with a '**Direct Integration**', while a Server is represented with an '**API Integration**'. Since we would like to inform users on events which are recorded by real-life PLCs or Servers which hold this information within them as data-points, these data-points must be defined within the Dashboard environment as **Variables** and **Triggers**. Variables are used to store informative data such as measurements, important values and similar, while Triggers are used to indicate that an event, worth notifying Users about, has happened. In case of a Direct Integration, the platform is responsible for getting data-points from the PLC. This is done once every minute. Data-point values are transferred using an .htm file which is generated by the platform and needs to be placed on the PLC. Figure 1.1 shows the corresponding data-flow with numbered purple lines. In case of an API Integration, when a real-life event happens, data-points (Trigger and Variable values) are sent to the platform via an API request. Figure 1.1 shows the corresponding data-flow with numbered orange lines.

For time dependent processes which are irrelevant of real-life equipment such as scheduled tasks or reminders on daily routines, so called **Time-Triggers** can be defined within the Dashboard environment.

So far, the Dashboard holds information on important events in form of defined **Triggers** and additional information on these events in form of defined **Variables**. Added **Users** candidate as notification receivers. The missing pieces are the notifications themselves and rules which define which users should get which notifications and when. Within the Dashboard environment, notifications and their content are defined using **Templates**. **Routing rules** are used to define which templates are to be sent to which users when a defined Trigger is triggered.

In Figure 1.1, the basic data flow is displayed with dark colored lines. The red lines indicate triggering of defined Triggers. When a Trigger is triggered, the platform combines the knowledge on defined Routing rules, Integrations and Users and generates notifications which are routed to real-life users according to the configured Routing rules. This way, some users will get only app notifications, some might get Email and SMS notifications, while some might not be notified at all.

The following sections give a more detailed introduction on the terminology used within the Herald Dashboard Environment.



## 1.1.2 Organizations

In the Herald Dashboard environment, the main subject to which nearly all later configuration is linked is an **Organization**. An organization is a virtual representation of the company for which the user is configuring the notification platform through Herald Dashboard.

Once everything is setup, employees of this company will become Herald users and they will receive Herald notifications about events in the Organization.

## 1.1.3 Groups

Nearly every company has sectors or divisions where different people are responsible for executing different tasks and have different duties. Thus, not every person in the company should be informed or bothered with notifications about events that are out of his/her area of interest or responsibility.

For this reason, for every Herald Organization, the administrator user can define **Groups**. Groups are virtual representations of the company's inner structure and can be used to group users by their responsibilities and duties, their roles within the company or even areas of activity. So, once everything is setup, the company's employees will become Herald users who are assigned to Groups.

Assignment to groups is not mandatory. Users can be assigned to one, multiple or no Group at all.

## 1.1.4 Users & Roles

Employees of the company which are supposed to get notified by the platform must first become Herald users within the corresponding Herald Organization.

The creator of an Organization automatically becomes the Organization's first User and this User has the administrative rights for this Organization. The administrator User can invite other users, define their roles and assignment to Groups. Every user must have a defined **Role**. Three role types are available:

- Administrator,
- Operator and
- Regular User.

Regular Users can only receive notifications within an Organization. Operator users are Regular Users with the additional permission to add, modify and delete Templates and Routing Rules. These users can also send Instant notifications. Administrator users are allowed to do all available configuration within an Organization.

Once everything is setup, employees of the company which are supposed to be notified by the platform will be defined as Herald Users within a Herald Organization and they will be assigned to Groups and given permissions according to the needs and preferences of their real-life company.

## 1.1.5 Integrations

Although Herald supports manually triggered notifications (1.1.10), it is important to understand that Herald is an automated notification platform which means that it is intended to be integrated with a source of events which result in notifications for Herald users. In Herald Dashboard, an **Integration** is the logical representation of such a source. Two different types of integrations are supported:

1. **Direct Integration**. This integration type allows the admin user to configure a connection to an S7-1x00 Siemens PLC (S7-1200 or S7-1500 Siemens CPUs with integrated web server, supporting user defined web pages).
2. **API Integration**. This integration type allows the admin user to configure a connection to the Herald API.



## 1.1.6 Triggers

In the Herald Dashboard environment, sending of notifications is initiated by modifying values of special variables called **Triggers**. In this document, the modification of these Trigger variables will often be referred to as triggering the Trigger. Three types of triggers can be defined:

1. **Direct Integration Triggers.** These triggers are linked to a previously defined Direct Integration. They are defined as links to the PLC's DB (data-block) data-points. If a DB data-point value change is detected, the corresponding Trigger is being triggered. The platform reads the values of all defined Triggers (DB data-point links), by analyzing the contents of a web page which is hosted on the PLC's internal web-server. This is done once every minute. The mentioned web page is basically an .htm file which is generated on the Herald Dashboard and needs to be downloaded to the PLC (read 2.9.3.4.2.5).
2. **API Integration Triggers.** These triggers are linked to a previously defined API integration. They are triggered from the client side via https requests.
3. **Time-triggers.** These triggers are not linked to an integration and as one would imagine, they are triggered by time. They can be used to trigger sending of daily or weekly notifications to remind users of time dependent events such as scheduled repetitive tasks, changes in shifts etc.

## 1.1.7 Variables

Notifications are usually used to report on changes in the client's environment. For example, the temperature has risen above a certain value, or the production rate has fallen below the targeted value. A notification is much more informative if it includes dynamic content with values which might be time or process dependent (current temperature or production rate).

In the Herald Dashboard environment, values from the client's environment which can change in time and are supposed to be dynamically inserted into notifications are called **Variables**. Variables are linked to previously defined integrations. Two types of variables can be defined:

1. **Direct Integration Variables.** These variables are linked to a previously defined Direct Integration. They are defined as links to the PLC's DB (data-block) data-points. The platform reads the values of all defined Variables (DB data-point links), by analyzing the contents of a web page which is hosted on the PLC's internal web-server. This is done once every minute. The mentioned web page is basically an .htm file which is generated on the Herald Dashboard and needs to be downloaded to the PLC (read 2.9.3.4.2.5).
2. **API Integration Variables.** These variables are linked to a previously defined API integration. Their values are provided along with the trigger from the client side via https requests.

## 1.1.8 Templates

Content of notifications which are to be sent to users is defined through **Templates**. Templates consist of a Title and Content. Depending on the type and severeness of the future message, Herald differentiates four type of messages. Each type is displayed in a different color in the Herald App:

- System Message (dark blue),
- Info Message (turquoise),
- Warning Message (orange) and
- Error Message (red).

Users can not send nor setup sending of system messages. These are messages sent from the Herald system to the Users (for example, invitations to join an organization etc.). Since templates are a generalized form of future notifications, a severeness level must be assigned to every defined template. This means that every Template must be either an Info, Warning or an Error notification.





In the Variables section, it has been explained that Variables' values can dynamically be inserted into notification's content. For this reason, templates are not only linked to an Organization, but can also be linked to an Integration within the Organization. In this case, all Variables linked to the corresponding Integration can be used within the Template. When creating such a template, Variables are dragged and dropped into the Template's Content or Title in form of **Placeholders**. When a notification is generated, placeholders will be replaced with the value of the corresponding Variable at that moment. There are also **System Placeholders** like 'User first name', 'Organization name' etc. When the notification is generated, System Placeholders will be replaced with the user's actual first name, or the name of the defined Organization.

### 1.1.9 Routing Rules

**Routing Rules** are used to define which users are supposed to get which notifications when a previously defined Trigger has been triggered. Every Routing Rule is defined by these five elements:

1. Organization,
2. Trigger,
3. Template,
4. Notification Medium,
5. User selection rule.

Basically, within the chosen Organization, when a chosen Trigger gets triggered, send the chosen Template to all Users which fall under the defined User selection rule. User selection rules allow selection of all users, specific users or users assigned to Groups and/or User with specific Roles.

### 1.1.10 Instant Notifications

**Instant Notifications** are intended to be used to generate one-time notifications. For example, if one would like to notify specific users that they should take care of some tasks (for example, pick something up while they are nearby), one could do that by sending instant notifications to those users. Within an Organization, only users with the Administrator or Operator role are allowed to send Instant Notifications.



## 2 CONFIGURATION

### 2.1 Herald Dashboard web page

A user can land on the Herald Dashboard login page, simply by navigating to the Herald Dashboard website:

<https://www.heraldservices.net>

Since the user is not yet authorized, the user will automatically be redirected to the Login page (Figure 2.1 - Login page).

On this page, existing users can enter their credentials and log into their Herald Dashboard.

Existing users who have forgotten their password and need to reset it can do this by clicking on the 'Forgot password' link.

New users who want to create a Herald Dashboard account can do that by clicking on the 'Register' link.

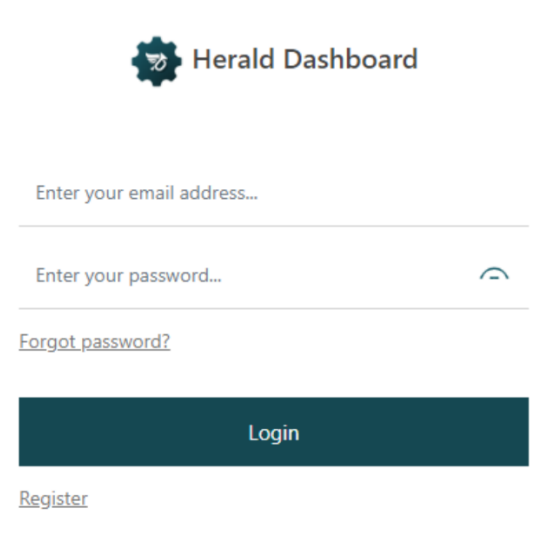


Figure 2.1 - Login page

### 2.2 Register new account

By clicking on the Register link below the Login button, the user is redirected to the register view.

Here, the user must enter a valid email address, a password and a password confirmation (reenter the same password).

The password must be 8-64 characters long, contain letters and contain at least one of the following: a digit, a combination of uppercase and lowercase letters, a special character.

After a click on the 'Register' button, the user should get an email with an activation link to his entered email account. By clicking on the link, the user confirms his identity and is redirected to the login page where he/she can enter his/her credentials and log into Herald Dashboard.

### 2.3 Forgot password

In case the user has forgotten his/her Herald Dashboard password, from the Login View page, the user can click on the 'Forgot password' link.

This will redirect the user to the Forgot password view where he/she can enter the email address associated with the corresponding Herald Dashboard account.

Once the email address is entered, the user can request a password change by clicking on the 'Request password reset' button. This will send an email with a password reset link which will redirect the user to a View where he/she will be required to enter and confirm the new password.



## 2.4 Login

Previously registered users can simply login from the login page by entering credentials and clicking on the Login button.

Once logged in, the first thing the user sees is our 'Privacy policy' and 'Terms and conditions statement'. The user must accept this to continue using Herald Dashboard. If the user should chose to decline, he/she will automatically be logged out. Users who accept the conditions and policies will arrive at the Herald Dashboard home view. Figure 2.2 shows the home view with numbers highlighting important sections:

1. At the home view, the user gets greeted with a welcome message.
2. Since this is the first time the user has logged in, no personal data is set. As long as this is true, the home view will offer a link for setting the user's personal data.
3. The left navigation pane allows the user to navigate through different sections of the Herald Dashboard application.
4. The user actions dropdown allows to user to navigate to the User settings view or to Log out.
5. The notifications icon allows the user to expand the user notifications pane (comes in from the right side of the application). This pane will show invites and indicate other events that require the user's attention.
6. The user support dropdown allows the user to send an email support request or to copy a support phone number to the clipboard. Other available options will open the 'Instructions' and 'About us' popups.
7. Link to the latest version of this document (Configuration manual).
8. The Herald dashboard home icon allows the user to navigate to the home view from all parts of the application.

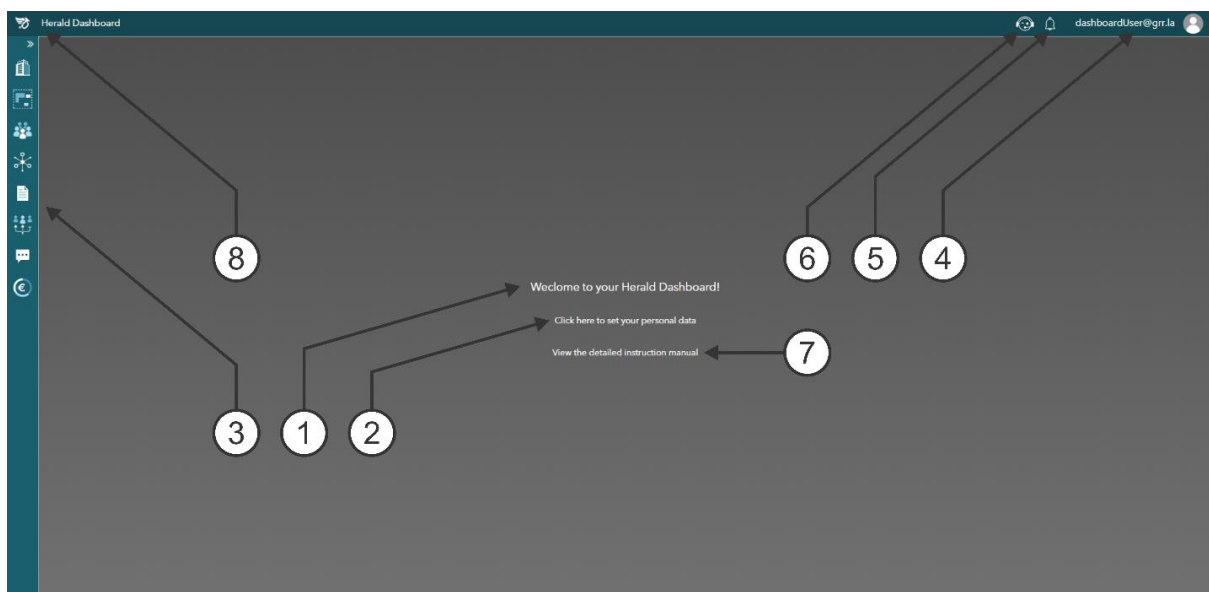


Figure 2.2 - Home View



## 2.5 Set personal data

As long as no user personal data is set, the home view will offer a link for setting the user's personal data (read 2.4, numbered item 2). This section can always be accessed from the User actions dropdown (read 2.4, numbered item 4), User settings option.

Figure 2.3 shows the User settings section. Here, the user can set his/her:

- First Name,
- Last Name,
- Email Address and
- Phone Number (must be specified in accordance to the [E.123](#) international standard).

The user can only view his/her Username (the email address assigned to the user's account). If you should need to change your Username, please contact our [support](#).

To change his/her password, the user must enter the current password, enter the new password and confirm the new password by reentering it. The new password must satisfy the constraints as described in section 2.2.

Figure 2.3 - User settings

Once a change to any of the editable fields is detected, the Save button will become enabled. By clicking on the enabled Save button, all changes are processed and persisted.

## 2.6 Organizations

### 2.6.1 About

All pages accessed from the left navigation pane (read 2.4, numbered item 3) except the Subscriptions page are organized in three sections:

1. Add / Refresh section,
2. Filter section,
3. Overview section.

Figure 2.4 shows the Organizations overview page with the previously described sections. By default, all currently defined organizations will appear in the Overview section. Organizations can be filtered by editing the Filter section.

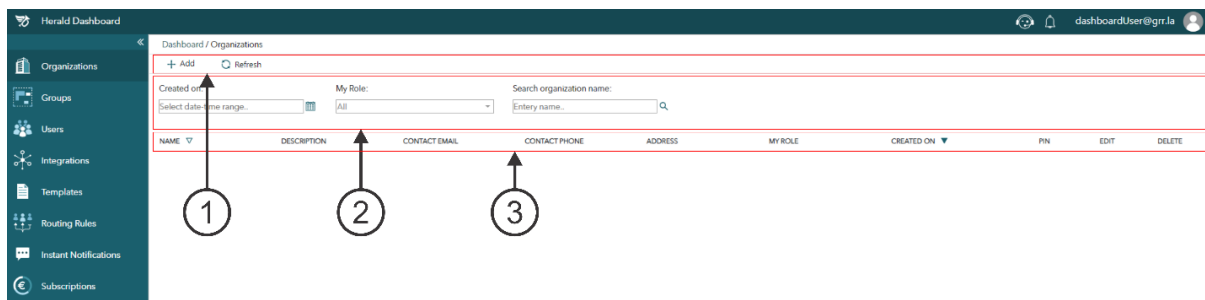


Figure 2.4 - Organizations overview

By clicking the Refresh button in the Add/Refresh section, the Overview section is reloaded with the currently set filter values. Organizations can be filtered by their creation date (range date picker) and/or by the user's role within the Organization (All roles, Administrator, Operator, Regular user). Filtered Organizations can be searched by their name. These options can be found in the Filter section.

In the overview section, Organizations are displayed in a table where each Organization is represented by one table row. Most of the table columns are dedicated for presenting the Organization specific data (entered by the Organization creator or Administrator user). The last three columns on the right side of the table are so called action columns. By clicking on the icon in the column, the user initiates an action for the Organization in the corresponding table row. This way, Organizations can be Pinned, Edited or Deleted.

### 2.6.2 Add new Organization

By clicking the Add button in the Add/Refresh section, the Add new organization dialog appears (Figure 2.5).

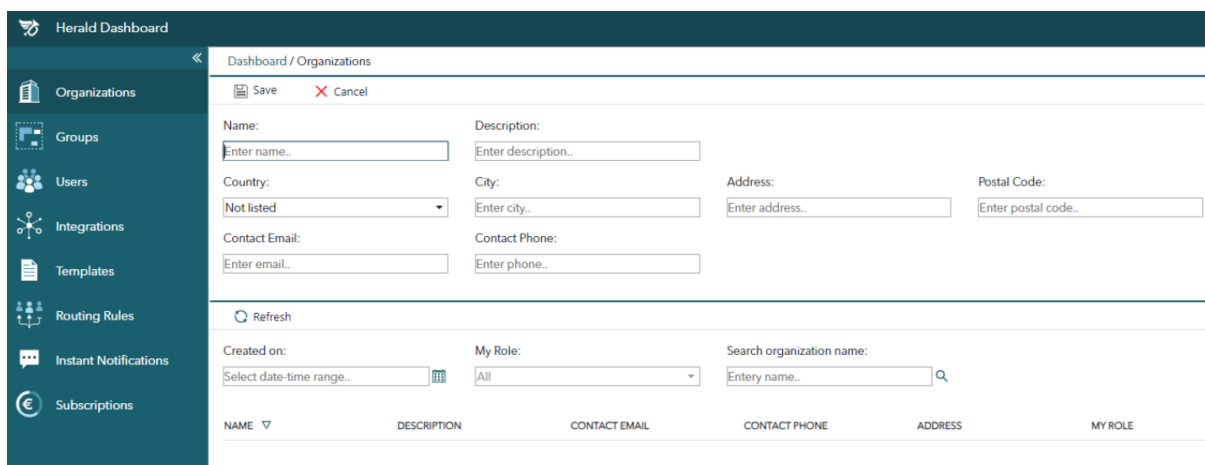


Figure 2.5 - Add new organization dialog



The Save icon is disabled as long as values are not entered into mandatory fields. In this case, Organization Name and Contact Email address are mandatory fields.

Users are not allowed to add multiple Organizations with the same name.

Even though one can save an Organization with values entered only for mandatory fields, we strongly suggest that all values are entered since they are valuable for further setup and support. Once mandatory fields have values, the Save button becomes enabled. By clicking on the Save button, the Organization gets added.

Figure 2.6 shows the overview section with one created Organization.

NAME	DESCRIPTION	CONTACT EMAIL	CONTACT PHONE	ADDRESS	MY ROLE	CREATED ON	PIN	EDIT	DELETE
Devoted	Herald creator	info@devoted.ba	+3873311111	The organization's address ...	Administrator	28.09.2019 18:52			

Figure 2.6 – Overview section with one created Organization

As already stated in 1.1.2, In the Herald Dashboard environment, the main subject to which nearly all later configuration is linked is an **Organization**. All billing related topics are also linked to the Organization. To understand the billing model and prices, please read section 3 and our [Pricing Policy Documentation](#).

### 2.6.3 Pin Organization

Figure 2.7 shows the Organizations overview with a single pinned Organization.

Imagine a single user having multiple Organizations defined. In this case, other overviews such as Groups, Users, Integrations etc. can easily become confusing which increases the chance of mistakes while configuring the Herald system for one of the Organizations.

For this reason, such Users, especially in the configuration phase, can pin the Organization they are currently configuring and all Organization selection dropdowns throughout the entire Herald Dashboard will be preset and locked to the pinned Organization and its related entities.

Now obviously, pinning the only available Organization as displayed in Figure 2.7 doesn't make much sense, but here the focus is on the reader learning the user experience when Organizations are pinned. A pinned Organization is easily unpinned by clicking on the same icon that was used to pin it.

If an Organization is pinned, the corresponding icon changes from gray to petrol color. Also, in the left navigation pane, the Organizations section appears with an orange pin, reminding the user that all configuration tabs are affected by the currently pinned Organization.

The pinned Organization will be persisted even after the user logs out.

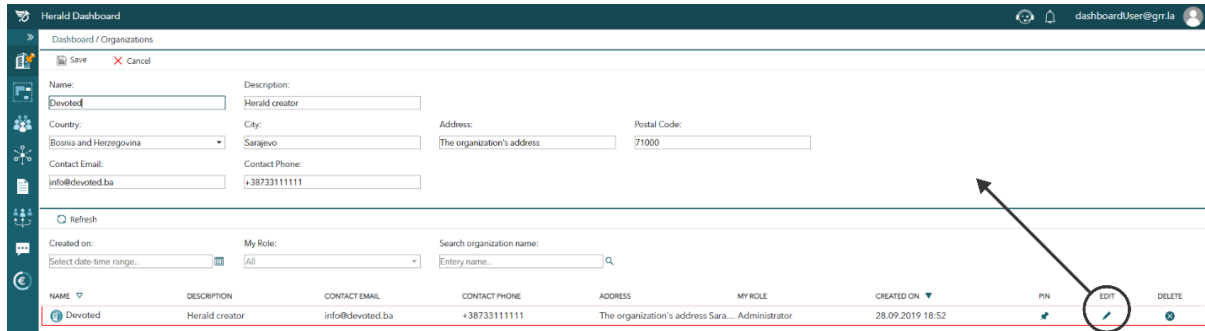
NAME	DESCRIPTION	CONTACT EMAIL	CONTACT PHONE	ADDRESS	MY ROLE	CREATED ON	PIN	EDIT	DELETE
Devoted	Herald creator	info@devoted.ba	+3873311111	The organization's address ...	Administrator	28.09.2019 18:52			

Figure 2.7 - Pinned Organization

## 2.6.4 Edit Organization

By clicking on the button in the Edit column, the corresponding Organization will be loaded into the Edit dialog for editing. If the user has insufficient permissions to edit an Organization, the edit button for the corresponding organization will be disabled. Only administrator users can edit an Organization.

Figure 2.8 shows the edit Organization dialog for the example Organization “Devoted”. Mandatory fields cannot be edited to be empty. The Save button becomes enabled once a change to any of the fields’ values is detected. At any time, the editing can be canceled by clicking on the Cancel button.



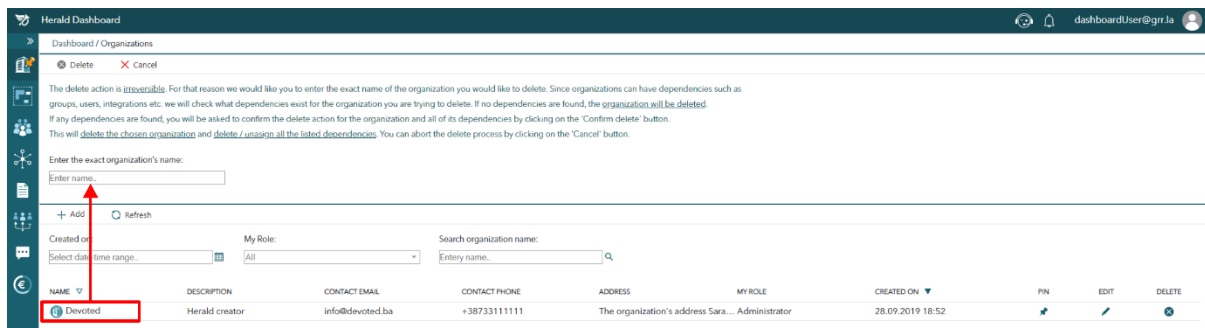
NAME	DESCRIPTION	CONTACT EMAIL	CONTACT PHONE	ADDRESS	MY ROLE	CREATED ON	PIN	EDIT	DELETE
Devoted	Herald creator	info@devoted.ba	+38733111111	The organization's address Sara...	Administrator	28.09.2019 18:52			

Figure 2.8 - Edit Organization dialog

## 2.6.5 Delete Organization

By clicking on the button in the Delete column, Herald services will initiate the deletion of the corresponding Organization. If the user has insufficient permissions to delete an Organization, the delete button for the corresponding organization will be disabled. Only administrator users can delete an Organization.

**The delete action is irreversible!** For that reason the user is required to enter the exact name of the Organization he/she would like to delete. Figure 2.9 shows that first step of the Organization deletion process.



The delete action is **irreversible**. For that reason we would like you to enter the exact name of the organization you would like to delete. Since organizations can have dependencies such as groups, users, integrations etc. we will check what dependencies exist for the organization you are trying to delete. If no dependencies are found, the organization will be deleted. If any dependencies are found, you will be asked to confirm the delete action for the organization and all of its dependencies by clicking on the 'Confirm delete' button. This will delete the chosen organization and delete / unassign all the listed dependencies. You can abort the delete process by clicking on the 'Cancel' button.

Enter the exact organization's name:

Enter name...

NAME	DESCRIPTION	CONTACT EMAIL	CONTACT PHONE	ADDRESS	MY ROLE	CREATED ON	PIN	EDIT	DELETE
Devoted	Herald creator	info@devoted.ba	+38733111111	The organization's address Sara...	Administrator	28.09.2019 18:52			

Figure 2.9 - Delete Organization dialog

Once the exact name of the Organization is entered, the Delete button becomes available. By clicking on the delete button, the user initiates the deletion process. Since Organizations can have dependencies such as Groups, Users, Integrations, overdue payments etc. the system will check what dependencies exist for the Organization the user is trying to delete. If no dependencies are found, the Organization will be deleted right away. If any dependencies are found, the user will be asked to confirm the delete action for the Organization and all of its dependencies by clicking on the 'Confirm delete' button. This will delete the chosen Organization and delete / unassign all the listed dependencies. The delete action can be aborted by clicking on the 'Cancel' button.

## 2.7 Groups

### 2.7.1 About

All pages accessed from the left navigation pane (read 2.4, numbered item 3) except the Subscriptions page are organized in three sections:

1. Add / Refresh section,
2. Filter section,
3. Overview section.

Figure 2.10 shows the Groups overview page with the previously described sections. By default, all currently defined Groups will appear in the Overview section. Groups can be filtered by editing the Filter section.

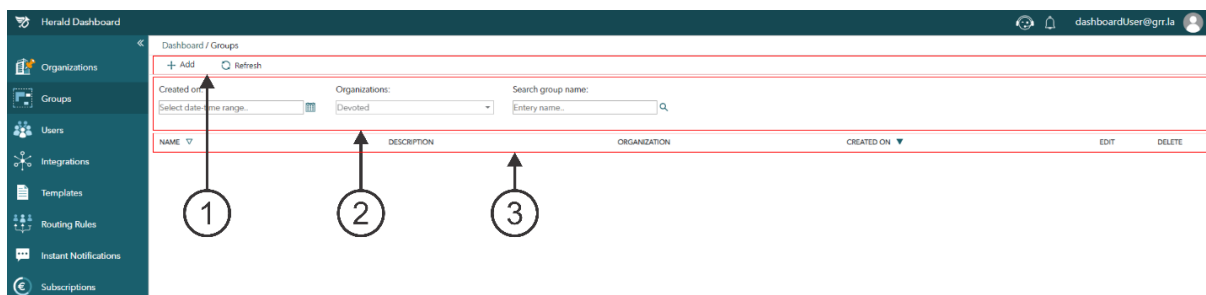


Figure 2.10 - Groups Overview

By clicking the Refresh button in the Add/Refresh section, the Overview section is reloaded with the currently set filter values. Groups can be filtered by their creation date (range date picker) and/or by their assignments to Organizations. Filtered Groups can be searched by their name. These options can be found in the Filter section.

In the overview section, Groups are displayed in a table where each Group is represented by one table row. Most of the table columns are dedicated for presenting the Group specific data (entered by the Group creator / editor). The last two columns on the right side of the table are so called action columns. By clicking on the icon in the column, the user initiates an action for the Group in the corresponding table row. This way, Groups can be Edited or Deleted.

### 2.7.2 Add new Group

By clicking the Add button in the Add/Refresh section, the Add new Group dialog appears (Figure 2.11).

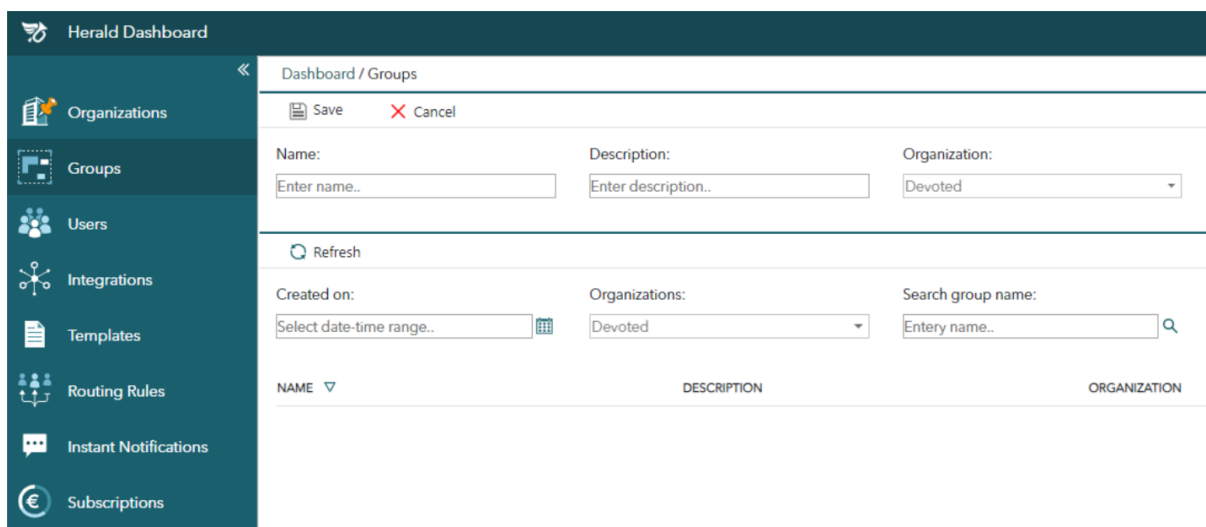


Figure 2.11 - Add new Group dialog





The Save icon is disabled as long as values are not entered into mandatory fields. In this case, Group name is the only mandatory field. It must be unique within an Organization. The Organization selection dropdown is preset to one of the previously added Organizations or, as in this case, to the pinned Organization. As previously explained, if an Organization has been pinned, all Organization selection dropdown throughout the entire Dashboard will be preset to this Organization and disabled for editing. The Group will be assigned to the Organization set in the Organization selection dropdown. Even though one can save a Group with values entered only for mandatory fields, we strongly suggest that all values are entered since they are valuable for further setup and support. Once mandatory fields have values, the Save button becomes enabled. By clicking on the Save button, the user Saves the Group.

Figure 2.12 shows the overview section with one created Group.

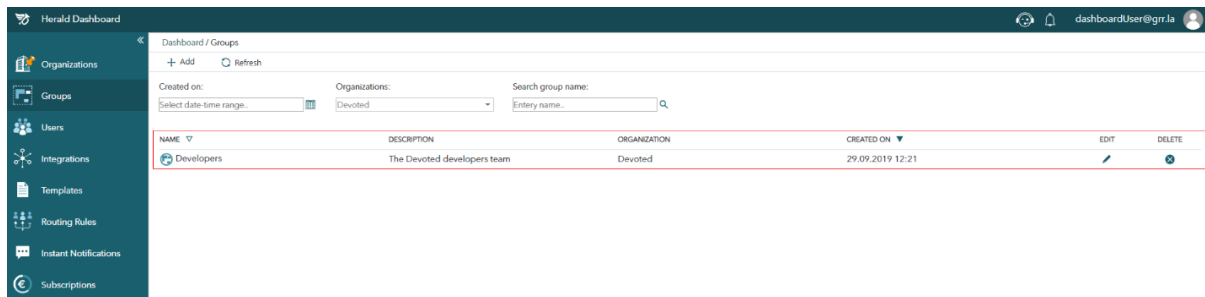


Figure 2.12 – Overview section with one created Group

### 2.7.3 Edit Group

By clicking on the button in the Edit column, the corresponding Group will be loaded into the Edit dialog for editing. If the user has insufficient permissions to edit a Group, the edit button for the corresponding Group will be disabled. Only administrator users can edit Groups.

Figure 2.13 shows the edit Group dialog for the example Group “Developers” within the “Devoted” Organization. Mandatory fields cannot be edited to be empty. The Save button becomes enabled once a change to any of the fields’ values is detected. At any time, the editing can be canceled by clicking on the Cancel button. The Group’s assignment to an Organization cannot be changed (to do this, delete the first Group, and add another one to the other Organization).

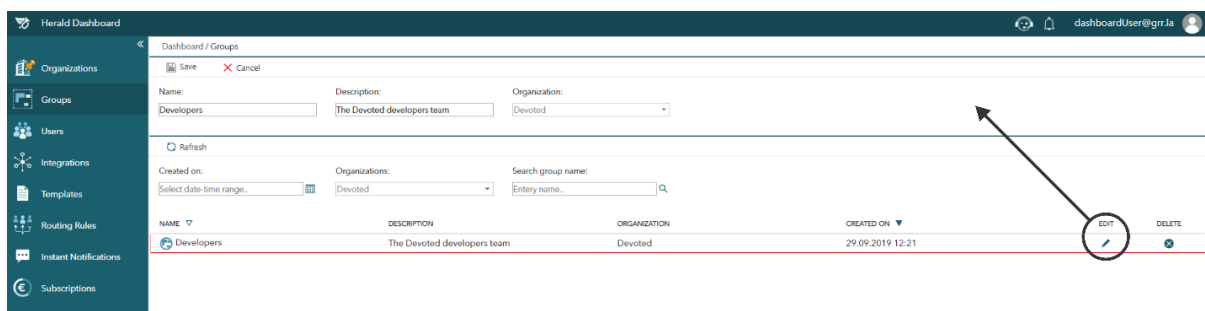


Figure 2.13 - Edit Group dialog



## 2.7.4 Delete Group

By clicking on the button in the Delete column, Herald services will initiate the deletion of the corresponding Group. If the user has insufficient permissions to delete a Group, the delete button for the corresponding Group will be disabled. Only administrator users can delete Groups.

**The delete action is irreversible!** For that reason the user is required to enter the exact name of the Group he/she would like to delete.

Figure 2.14 shows that first step of the Group deletion process.

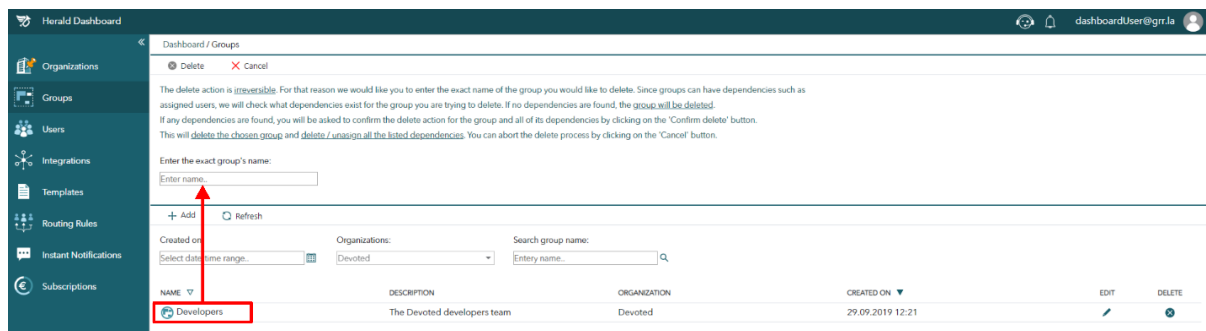


Figure 2.14 - Delete Group dialog

Once the exact name of the Group is entered, the Delete button becomes available. By clicking on the delete button, the user initiates the deletion process. Since Groups can have dependencies such as assigned Users, the system will check what dependencies exist for the Group the user is trying to delete. If no dependencies are found, the Group will be deleted right away. If any dependencies are found, the user will be asked to confirm the delete action for the Group and all of its dependencies by clicking on the 'Confirm delete' button. This will delete the chosen Group and unassign all listed dependencies. The delete action can be aborted by clicking on the 'Cancel' button.

## 2.8 Users

### 2.8.1 About

All pages accessed from the left navigation pane (read 2.4, numbered item 3) except the Subscriptions page are organized in three sections:

1. Add / Refresh section,
2. Filter section,
3. Overview section.

Figure 2.15 shows the Users overview page with the previously described sections. All users which have completed the registration process either through the Herald Dashboard or via the mobile app are considered Herald users. To allow any Herald user to utilize the services of the Herald Platform (for example receive notifications), the corresponding user must first be assigned to a Herald Organization. This can be done in two ways:

- The user creates an Organization and automatically gets assigned to it or
- The user accepts an invite to join an Organization created by other users.

By default, all currently defined Users which have been assigned to the available Organizations will appear in the Overview section. Every Organization has at least one user (the organization creator). Users can be filtered by editing the Filter section.

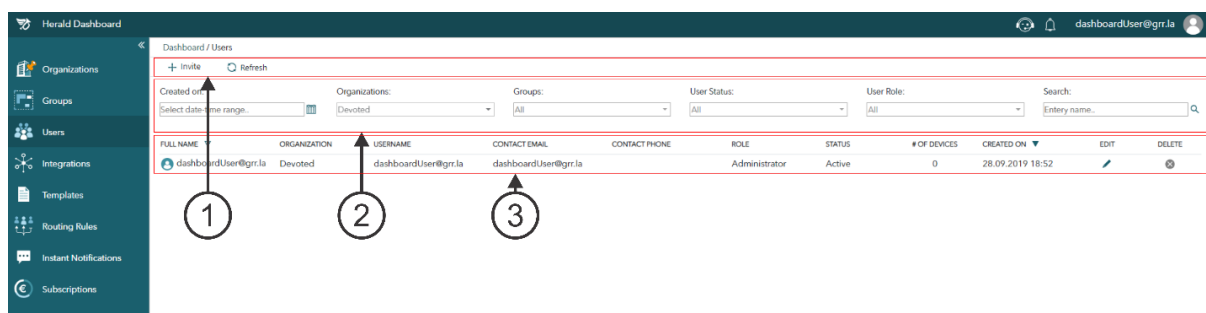


Figure 2.15 - Users overview

By clicking the Refresh button in the Add/Refresh section, the Overview section is reloaded with the currently set filter values. Users can be filtered by their creation date (range date picker), assignment to an Organization, assignment to Groups, their Status and Role. Filtered Users can be searched by their name and username. These options can be found in the Filter section.

In the overview section, Users are displayed in a table where each User is represented by one table row. Most of the table columns are dedicated for presenting the User specific data (entered by the User creator or Administrator user). The last two columns on the right side of the table are so called action columns. By clicking on the icon in the column, the user initiates an action for the User in the corresponding table row. This way, Users can be Edited or Deleted.

### 2.8.2 User Roles

Herald Users can have one of three Roles:

- Administrator,
- Operator,
- Regular User.

The Organization's creator is by default an Administrator User within that Organization. This user has all the rights and is considered the responsible User for handling the Herald Dashboard account of that Organization. Multiple Administrators are allowed, but one should be very careful when assigning the administrative role to Users.



A typical scenario would be that an engineer who is configuring the Herald system for a client, once finished and ready to hand off the system to the client, invites the client's employee who will be the responsible person for Herald as the new Administrator user. Once the new user familiarizes himself/herself with Herald Dashboard, he/she can remove administrative rights or even delete the engineer's user. From that moment on, the new responsible person for the Organization's Herald account will be the client's employee.

An Operator has less rights than the Administrator. This user can add/edit/delete Notification Templates, Routing Rules and send Instant Notifications. Regular Users can only view the Organization's setup but they cannot do any configuration within the Organization what so ever.

Generally speaking, users with the Administrator Role will most probably be actively using the Herald Dashboard since they are responsible for all configuration within the Organization and for making payments to the Organization's account. These users should have the Herald app installed since this is a way for them to confirm that their setup is working as intended. All administrator user should read and understand the [Pricing Policy Documentation](#).

Operator Users will also most probably be actively using the Herald Dashboard. They cannot make payments nor can they do most of the configuration within the Organization, but some tasks like adding/editing/deleting templates or routing rules will require them to log into Herald Dashboard from time to time. These users should have the Herald app installed since this is a way for them to confirm that the changes they have made have the intended effect.

Regular Users will rarely need to use the Herald Dashboard since they cannot make payments nor can they do any of the configuration within the Organization. However, they can use the dashboard to set their own user data and view all non-sensitive Organization data. These users will be using Herald mainly through the mobile app.

## 2.8.3 Invite / Add Users

By clicking the Invite button in the Add/Refresh section, the Invite new User dialog appears (Figure 2.16).

FULL NAME	ORGANIZATION	USERNAME	CONTACT EMAIL	CONTACT PHONE	ROLE	STATUS	# OF DEVICES	CREATED ON	EDIT	DELETE
dashboardUser@grn.la	Devoted	dashboardUser@grn.la	dashboardUser@grn.la		Administrator	Active	0	28.09.2019 18:52		

Figure 2.16 - Invite new User dialog

The Invite icon is disabled as long as values are not entered into mandatory fields. In this case, the new User's email address is the only mandatory field. Even though one can invite a User with values entered only for mandatory fields, we strongly suggest that all values are entered since they are valuable for further setup and support. This is especially important for inviting new Users since the Organization administrator should want to make sure that the new User is invited with an assignment to **fitting Groups** and an **appropriate Role**. Also, user data such as contact Email and Contact Phone Number can be very important if the user to be invited is supposed to receive notifications via SMS or Email.

**If Contact Email and Contact Phone Number are not populated by the inviter and the invitee does not enter this data upon registration, receiving of SMS or Email notifications will not be possible for the corresponding user.**

Once mandatory fields have values, the Invite button becomes enabled. By clicking on the Invite button, the new user is added to the corresponding Organization. The new user's status will be "Invited". For the user to become



an “Active” user, he/she must accept the invitation. Only active users can receive notifications. The invited user should get an invitation email to the configured email address. For Users new to Herald, the mail should offer instructions on registration options, either via the mobile app, or through the Herald Dashboard website.

We suggest that users with the role of Regular User complete their registration through the mobile app since they will most likely never need to access their Dashboard account. Administrator and Operator users can complete their registration however it suits them (through the mobile app, or online through Herald Dashboard).

In case the registration has been completed via the mobile app, after logging into the app the user should get an invitation notification and a welcome message. Figure 2.17 shows an example of this scenario. By clicking on the accept invite button, the user accepts the invitation via the Herald mobile app and becomes an “Active” user within the Organization.

In case the registration has been completed via Herald Dashboard, after logging in, the user should land on the Home View. In the upper right section, the notification icon should have a ringing animation with a number indicating the notifications count above it (read 2.4, numbered item 5). Figure 2.18 shows the notification icon. Figure 2.19 shows the expanded notification pane with one invitation notification present. By clicking on the notification, the user accepts the invitation. This way, the user becomes an “Active” user within the Organization.

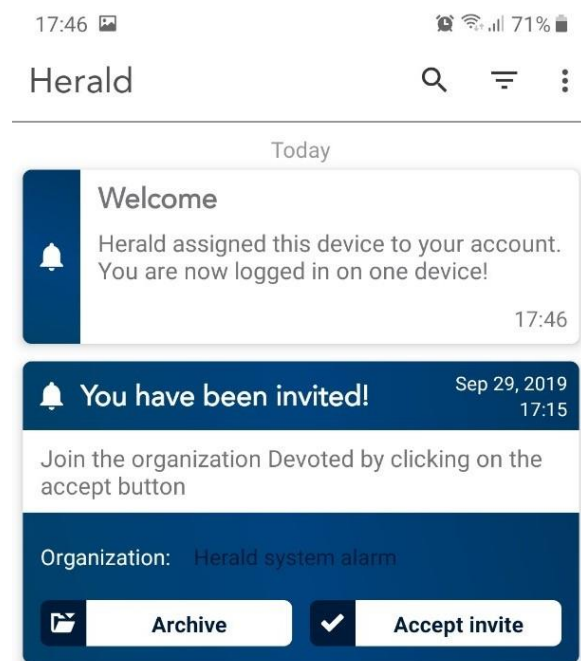


Figure 2.18 - Notification signalization

A click on the notification bell expands the right-side notification panel. Invites to organizations and other important info will appear in this section.

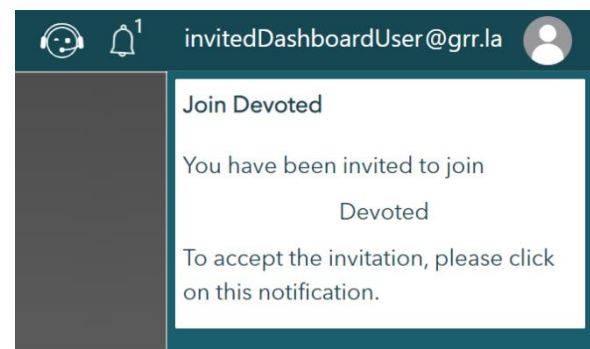


Figure 2.17 - Accept invitation via App

Figure 2.19 - Accept Invitation via Dashboard

Figure 2.20 shows the overview section after the invited User has accepted the invitation.

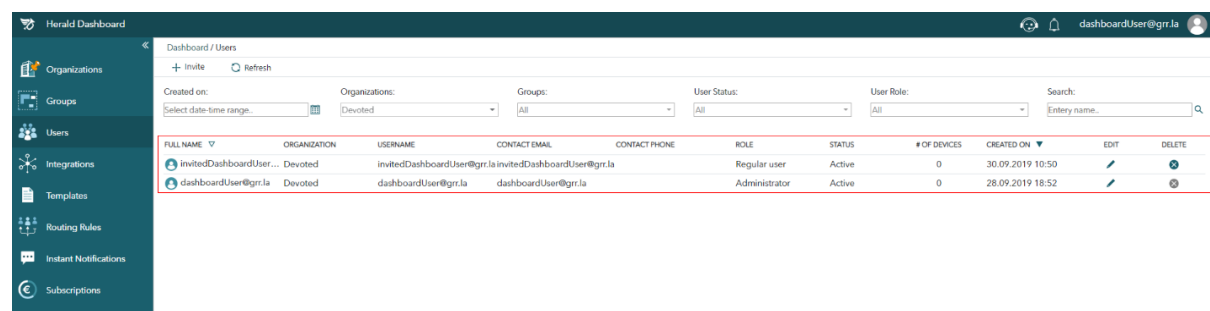


Figure 2.20 – Overview section with one created User



## 2.8.3.1 Subscription handling

All newly added Users are automatically assigned to the preexisting users subscription. By default, every new Organization gets a **Basic Users Subscription** which supports up to 5 added or invited users for free. Every time a user is invited the counter of assigned users to this subscription is increased. When a user is deleted, this counter is decreased.

To be able to invite/add more than 5 users, the subscription must be upgraded to the **Consumption users Subscription**. The pricing for the Consumption Users Subscription can be viewed in the [Pricing Policy Documentation](#).

## 2.8.4 Edit User

By clicking on the button in the Edit column, the corresponding User will be loaded into the Edit dialog for editing. If the user has insufficient permissions to edit a User, the edit button for the corresponding User will be disabled. Only administrator users can edit a User.

Figure 2.21 shows the edit User dialog for the administrator User within the example Organization “Devoted”. Mandatory fields cannot be edited to be empty. The Save button becomes enabled once a change to any of the fields’ values is detected. At any time, the editing can be canceled by clicking on the Cancel button.

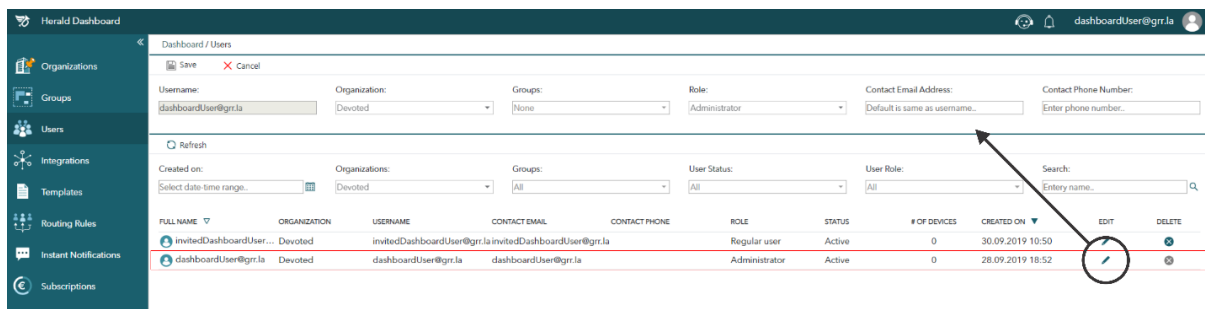


Figure 2.21 - Edit User dialog

## 2.8.5 Delete User

By clicking on the button in the Delete column, Herald services will initiate the deletion of the corresponding User. If the user has insufficient permissions to delete a User, the delete button for the corresponding User will be disabled. Only administrator users can delete a User.

**The delete action is irreversible!** For that reason the User is required to enter the exact name of the User he/she would like to delete. Figure 2.22 shows that first step of the User deletion process.

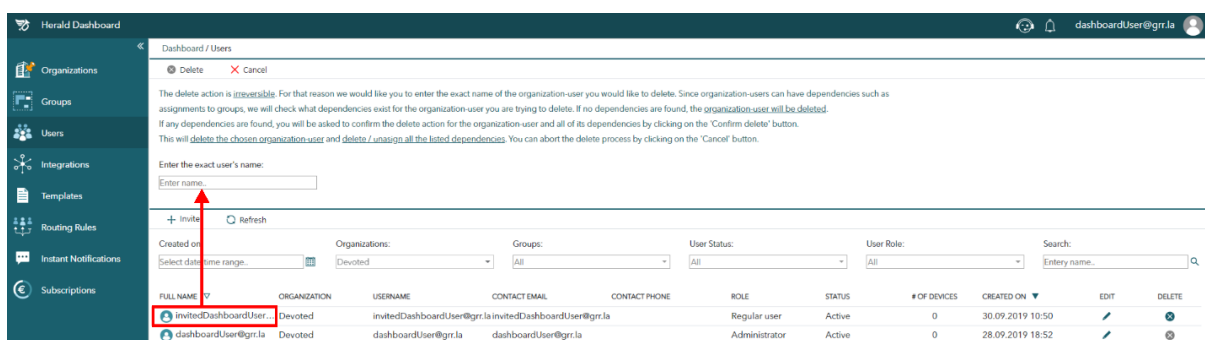


Figure 2.22 - Delete User dialog



Once the exact name of the User is entered, the Delete button becomes available. By clicking on the delete button, the user initiates the deletion process. Since Users can have dependencies such as assignment to Groups, the system will check what dependencies exist for the User to be deleted. If no dependencies are found, the User will be deleted right away. If any dependencies are found, the User will be asked to confirm the delete action for the User and all of its dependencies by clicking on the 'Confirm delete' button. This will delete the chosen User and delete / unassign all the listed dependencies. The delete action can be aborted by clicking on the 'Cancel' button.



## 2.9 Integrations

### 2.9.1 About

Herald Dashboard is a configuration tool which allows users to configure automated delivery of notifications on real life events. In the Dashboard environment, an **Integration** is the logical representation for a source of such events (a device on the client's side that the Herald service can communicate with to find out if such events have happened).

Two types of integrations are supported:

- Siemens S7-1200 or S7-1500 PLCs with integrated web server and support for user defined web pages,
- Any device (IoT, Server, Machine) or Computer program (SCADA like WinCC or other, business software solutions etc.) capable of sending http(s) requests to our API.

An integration with a PLC device is represented with a '**Direct Integration**', while an integration which communicates with our API is represented with an '**API Integration**'. Since we would like to inform users on events which are recorded by real-life PLCs or Servers which hold this information within them as data-points, these data-points must be defined within the Dashboard environment as **Variables** and **Triggers**. Variables are used to store informative data such as measurements, important values and similar, while Triggers are used to indicate that an event, worth notifying Users about, has happened (for example, we would consider a Trigger being triggered when a measurement is above a limit value, while the exact measurement at that point in time or the count of times this has happened in the last 24 hours would be stored in a Variable).

In case of a Direct Integration, the platform is responsible for getting data-points from the PLC. This is done once every minute. Data-point values are transferred using an .htm file which is generated by the platform and needs to be placed on the PLC (read 2.9.7). Figure 1.1 shows the corresponding data-flow with numbered purple lines. The first line represents the platform controlling the minute based timer. Each minute, a request is sent to the PLC, asking for the .htm file which holds information about the current values of Triggers and Variables. The PLC's inner web server returns the page as a response. This is represented with lines 2 and 3. The cyclic process evaluates the received response and reads the defined Triggers' and Variables' current values. These are then temporarily stored within the platform so that the next cycle's response can be compared to them. This is represented with lines 4 and 5. If a Trigger value change compared to the previously recorded response is detected, this Trigger is considered to be triggered.

In case of an API Integration, when a real-life event happens, data-points (Trigger and Variable values) are sent to the platform via an API request. Figure 1.1 shows the corresponding data-flow with numbered orange lines.

Herald offers time dependent triggers, **Time-triggers**. These can be used to trigger sending of notifications on scheduled events which are not related to any Integration.

To allow for easier configuration and management of Integrations, Time-triggers, Triggers and Variables, the left navigation pane's (read 2.4, numbered item 3) Integration section has its own top navigation pane (view 2.9.2, numbered item 4) to navigate between the four subsections.

### 2.9.2 Integrations overview

All pages accessed from the left navigation pane (read 2.4, numbered item 3) except the Subscriptions page are organized in three sections:

1. Add / Refresh section,
2. Filter section,
3. Overview section.

Figure 2.23 shows the Integrations overview page with the previously described sections. As you can see, the Integrations view has an additional section:

4. Integrations navigation section.



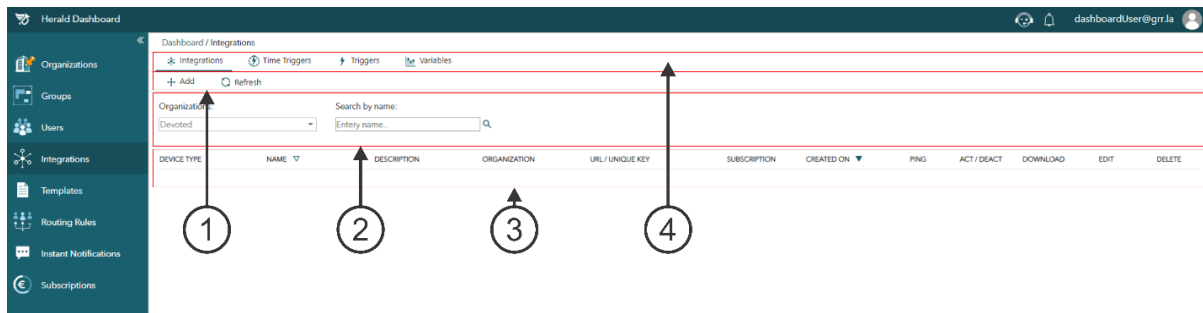


Figure 2.23 - Integrations Overview

The Integrations navigation section is used to navigate between different Integration-related configuration screens (Integration overview, Time-triggers overview, Triggers Overview and Variables overview) which will be explained in detail in the chapters to come.

By default, all currently defined Integrations will appear in the Overview section. Integrations can be filtered by editing the Filter section. By clicking the Refresh button in the Add/Refresh section, the Overview section is reloaded with the currently set filter values. Integrations can be filtered by their assignment to an Organization. Filtered Integrations can be searched by their name. These options can be found in the Filter section.

In the overview section, Integrations are displayed in a table where each Integration is represented by one table row. Most of the table columns are dedicated for presenting the Integration specific data (entered by the Integration creator / editor). The last five columns on the right side of the table are so called action columns. By clicking on the icon in the column, the user initiates an action for the Integration in the corresponding table row. This way, Integrations can be Pinged (2.9.5), Activated / Deactivated (2.9.6), Downloaded (2.9.7), Edited (2.9.8) or Deleted (2.9.9).

### 2.9.3 Add S7-1x00 (Direct) Integration

By clicking the Add button in the Add/Refresh section, the Add new Integration dialog appears (Figure 2.24).

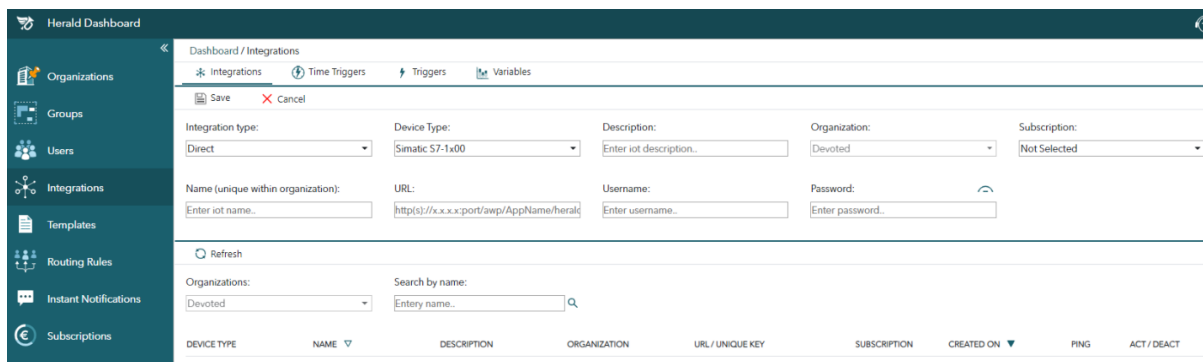


Figure 2.24 - Add new Integration dialog

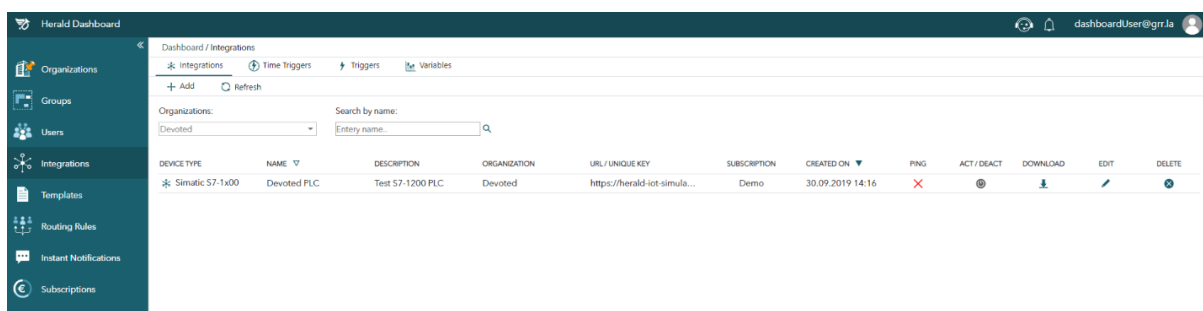


Figure 2.25 – Overview section with the added Direct Integration



The Integration type dropdown allows the User to select the Direct or API Integration type. Changes to this selector directly translate to changes in the Device Type dropdown (**Direct** to **Simatic S7-1x00**, **API** to **Server** and vice versa). Obviously, to add a Direct Integration, the **Direct** option needs to be selected.

The Save icon is disabled as long as values are not entered into mandatory fields. In this case, Integration name, URL, Username and Password are the mandatory fields. The Organization selection dropdown is preset to one of the previously added Organizations or, as in this case, to the pinned Organization. As previously explained, if an Organization has been pinned, all Organization selection dropdown throughout the entire Dashboard will be preset to this Organization and disabled for editing. The Integration will be assigned to the Organization set in the Organization selection dropdown. Even though one can save an Integration with values entered only for mandatory fields, we strongly suggest that all values are entered since they are valuable for further setup and support.

Once mandatory fields have values, the Save button becomes enabled. By clicking on the Save button, the user Saves the Integration. Figure 2.25 shows the overview section with the added Direct Integration.

### 2.9.3.1 Subscription selection

Generally speaking, all billing related matters are handled through Subscriptions (read 3). Unlike other services provided by the platform (API integration, SMS subscription etc.), where a single subscription covers all related expenses and thus can be managed from a single place (the Subscriptions section), every Direct integration requires a separate subscription to be able to fully operate so it can be viewed from the Subscriptions section but **is managed from the Integrations section only**. For this reason, when adding a Direct Integration, the Subscription selection dropdown becomes available. The user can choose between three options:

- Not Selected,
- Demo Direct Subscription,
- Classic Direct Subscription.

The first option will create a new Direct Integration which is not subscribed to any kind of subscription. **For the Integration to fully operate, it must be subscribed to a Subscription!** Subscriptions can be changed later through the Edit action.

The second option will assign the newly created Direct Integration to a preexisting Demo Direct Subscription. Every new Organization gets a **100% free Demo Direct Subscription** which can be used for testing purposes. Only one PLC can be assigned to the Demo subscription at a time. If an Integration had previously been assigned to the Demo Subscription, this dropdown option would not have been available. Under the Demo Direct Subscription, the Integration can be tested **for up to 12 ACTIVE hours** (from the overview section, Direct Integrations can be activated/deactivated - read 2.9.6). After this, the subscription will become usage deactivated. To prevent unintended usage-deactivation scenarios, the newly created Integration will be assigned to the Demo subscription in a **DEACTIVATED** state so that the user must manually activate it.

The third option will create a new **Classic Direct Subscription** and assign the newly created Direct Integration to this subscription. Under the Classic subscription, the Integration can work for as long as there are enough resources in the Organization's account (read 3.3.1 to learn how to add resources to your Organization's balance). The pricing for the Classic Direct subscription can be viewed in the [Pricing Policy Documentation](#). To prevent unintended costs, the newly created Integration will be assigned to the Classic subscription in a **DEACTIVATED** state so that the user must manually activate it.

The free hours (in case of a Demo Subscription) / paid usage (in case of a Classic Subscription) counters are **only active when the Integration is ACTIVATED!**



## 2.9.3.2 URL

To communicate with the PLC, Herald Services must have access and thus know the exact URL of the .htm file generated by the platform to be hosted on the PLC's web server. This value is entered into the URL field which has a placeholder to help the User enter the URL in a correct format:

**http://x.x.x.x:port/awp/AppName/heraldnotifications.htm**

The green part of the example URL is to be replaced with the exact public IP address of the device (including port if existing). The orange part of the example URL corresponds to the User-defined pages, Application name setting within the TIA Portal project (read 2.9.3.4.2). Since this Application name will be a part of the Integration URL, we strongly recommend using only standard URL supported characters (**do not use special characters, spaces and similar**). The remaining parts of the example URL do not have to be edited.

## 2.9.3.3 Username & Password

For Herald to be allowed to access the PLC's user defined web pages, the Integration's Username and Password must be defined. For this reason, a dedicated Herald user must be defined within the PLC's Web server's User management section (read 2.9.3.4.2). We suggest to add a Web server user with a Name containing no special characters nor spaces. The access level should include only the **"open user-defined web pages"** and **"write in user-defined web pages"** options (evaluates to a restricted access level).

Set the password to your preferences. The exact same Username and Password must be entered into the corresponding Integration definition fields.

Since user defined web-pages can be accessed locally, it is very important **not to use the user dedicated for Herald Services for local access!** The PLC's web server can handle only a limited number of sessions which can be used up due to user's not logging out. For the Herald dedicated user, Herald services are taking care of logouts, so to prevent any issues with accessing the PLC remotely with this user...

**...do not to use the user dedicated for Herald Services to directly access user defined web pages!**

When you are accessing the user defined web pages directly (via a browser) with a password protected account, make sure to...

**...log out as that user every time you finish work!**

Also, since the internal PLC's web server is handling the user authorization cookies and their validity using its internal time settings, it is important to...

**...keep the internal PLC time up to date!**

As you can see, to achieve a functional and reliable Direct Integration, some of the settings need to be done on the PLC side as well. This leads us to the next chapter which explains how to configure the S7-1x200 PLC.



## 2.9.3.4 Configure S7-1x00 PLC

To achieve a connection between the Herald service and your S7-1x00 PLC, the PLC must be configured appropriately.

In this section, an example scenario will be explained using the following PLC:

- Short designation: CPU 1212C DC/DC/Rly
- Article number: 6ES7 212-1HE40-0XB0
- Firmware version: V4.2

This manual assumes previous knowledge in using TIA Portal for PLC programming. Basic TIA Portal usage related topics will not be addressed.

### 2.9.3.4.1 Network setup

To achieve a connection between the Herald service and your S7-1x00 PLC, the PLC must have an **integrated web server** and support **user defined web pages**. To check whether your device meets these requirements, please read the device's manual or contact your local SIEMENS partner.

The PLC must be **connected to the internet**. Also, the modem through which the PLC is connected to the internet must have a **static public IP address**. To get a static IP address assigned to your modem, you must either:

- contact your Internet Service Provider (ISP) to acquire a public static IP (**PREFERRED SOLUTION**) or
- integrate with a dynamic DNS service provider (like [dynDNS](#)) to dynamically route the traffic as your public IP changes.

Once your modem has a static public IP address, it is reachable from all around the globe. Technically, the modem is responsible for the connection to the internet while a router connects the devices within your network to the modem. For Herald services to communicate with your PLC, your router needs to be configured to route the traffic from your public static IP to the PLC. If you are having trouble identifying your modem/router, the following few sentences could be helpful.

In case of separate modem/router configurations, the modem is usually the first device that connects to the cable provided by your ISP, while the router is usually the first device which is connected to the modem. Lately, both modem and router are delivered as a single combo-device which is responsible for both connecting to the internet, and routing the traffic to the internally connected devices. In this case, you might only be looking for one device. If you still can't identify your modem/router, contact your ISP to help you out.

To configure your router to route the traffic to your PLC, you must open a browser and access your router's console by entering the routers default IP address in the browser's address bar. For example, TP-LINK routers have the default IP address of **192.168.0.1** and **admin** as both the username and password.

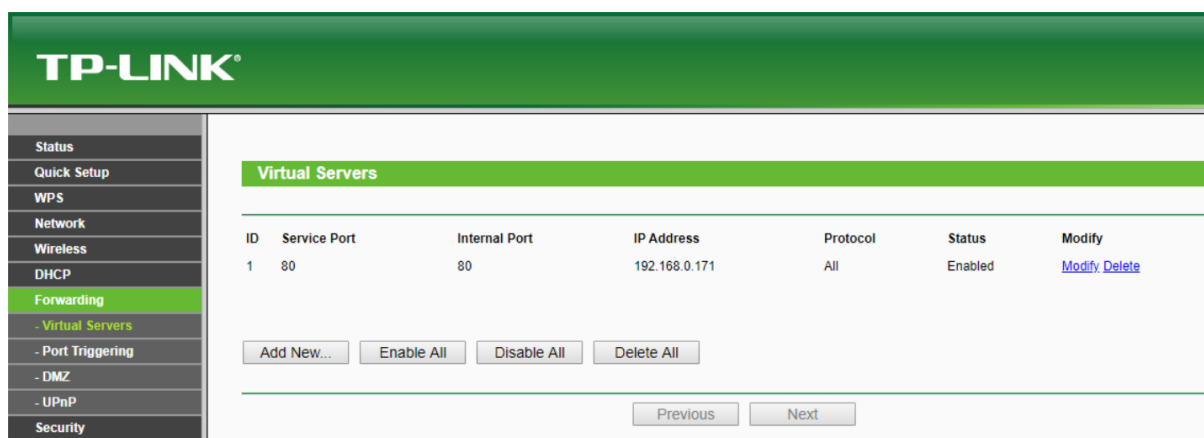


Figure 2.26 - Port Forwarding example



Once inside the router console, look for a **Forwarding** setup section, **Virtual Servers** subsection (slight terminology variations might occur depending on your router's manufacturer). Here, you will have to add a rule to forward traffic from the Service Port of your preference (in this case, we selected the default port 80) to the Internal service port 80 on the IP address of your PLC (as defined in your TIA Portal project, for example 192.168.0.171). Now, your network is configured to route communication from Herald services to your PLC.

The selected service port must be...

**...included in the Integration's URL definition (read 2.9.3.2)!**

If the default service port has been chosen (80)...

**...it can be left out of the Integration's URL definition (read 2.9.3.2)!**

### 2.9.3.4.2 TIA Portal setup

#### 2.9.3.4.2.1 Example Application

The following link offers a download of an example TIA Portal (Version V15.1 Update 3) application. For earlier versions of the example application, please contact our [support](#).

[Download example application](#)

The application has a preconfigured Device configuration and all necessary DBs and FBs defined.

Regarding the DBs, every TIA Portal application which is configured to communicate with Herald services needs to have following DBs defined:

- Trigger DB (**HeraldTriggers**, **DB9998** in the example application) – This DB will be surveilled by Herald services to check if an alarm has occurred. Each data-point within the DB is defined as an integer where its value represents the number of times the corresponding alarm has occurred. Only data-points which are defined as Direct Integration Triggers will be surveilled (**S7 DB Name** and **S7 Trigger Name** inputs are used to define the DB and data-point name in the 'Add Trigger' dialog...read 2.11.1.2). If a change to any of the surveilled DB's data-points is detected, the corresponding Trigger will be triggered.
- Trigger Acknowledgement DB (**HeraldAckTriggers**, **DB 9997** in the example application) – This DB is used to allow users to initiate actions on the PLC's side by acknowledging notifications from their mobile app. If a User acknowledges a notification which was generated by triggering a Trigger to which this acknowledgement data-point has been assigned (**S7 ACK DB Name** and **S7 ACK Trigger Name** inputs are used to define this DB and data-point name in the 'Add Trigger' dialog...read 2.11.1.2), Herald services will write (set the corresponding data-point to value True). Each data-point is defined as a Bool where its value can be used to automatically initiate actions once a notification has been acknowledged.
- Variable DB (**HeraldVariables**, **DB9999** in the example application) – This DB is used to store all process related values that need to be included into user notifications (Bool, Int and Real data-types are supported). Only data-points which are defined as Direct Integration Variables will be surveilled (**S7 DB Name** and **S7 Variable Name** inputs are used to define the DB and data-point name in the 'Add Variable' dialog...read 2.12.1.2).

Regarding the FBs, the "**TriggerSet FB**" is used to set a data-point in the surveilled **HeraldTriggers** DB. It basically makes sure that no overflow can happen to the internal DB's data-points in case of a large number of detected alarms from the process side. This is necessary since Herald services check for changes to the data-point's values.



The “**TriggerAck FB**” is used to make sure that a data-point which has already been set to value 1 due to an acknowledged notification is reset to 0 after a defined time (defined at the Reset Time input and set to 5s in this example). This FB also transfers the detected data-point change to its four in/out inputs (Action Set, Action Reset, Action Increment and Action Decrement) which will react according to their names when an acknowledgement data-point is set to value 1. If you don’t need them, they can be left unused.

In the Main OB1 block, the first network is for synchronizing the user pages. Its CTRL\_DB input corresponds with the DB control DB for the user defined web pages, configurable from the Device Configuration -> Web server -> User-defined pages section.

The second network contains an example setup for one process alarm (%M100.0) named “Your Alarm”. For example, this could be an indication that a measurement is above a certain value or any other process related indication. This information is provided to the “TriggerSet” FB instance as an input (“Alarm Trigger”) which will increment the “HeraldTriggers”.t1 data-point in case a positive edge on that input is detected. Each time a positive edge is detected, it will be reflected in the change of the corresponding HeraldTriggers DB data-point which will be read by Herald services. On the Platform side, if this data-point has been defined as a Trigger, the detected change will cause the triggering of that Trigger (can result in notifications depending on the defined Routing rules). If for that Trigger, an acknowledgement data-point has been defined as S7 ACK DB Name: “HeraldAckTrigger, S7 ACK Trigger Name: “t1”, a click on the notification’s acknowledge button (available on the mobile app) will set the HeraldAckTriggers data-point to the value 1. After the defined 5 seconds, this value will be reset to 0.

#### 2.9.3.4.2.2 Data-blocks Setup

For Herald services to communicate with your PLC you will have to create some DBs which will serve as proxy for the transfer of data between Herald services and the PLC.

One DB should be defined for all triggers. In this DB, for every process related alarm, you should define a data-point. The DB and data-point names are arbitrary but should not contain special characters nor spaces. For example, the DB could be named **HeraldTriggers**, and inside the DB, one can define as many integer data-points as one would like. For practicality reasons, we suggest that you name them **t1, t2, t3...** Use the **TriggerSet FB** (read 2.9.3.4.2.1) to connect your process alarms to this Data-block’s internal data-points.

One DB should be defined for all trigger acknowledgements. In this DB, for every trigger you would like to be able to acknowledge through the mobile app, you should define a data-point. The DB and data-point names are arbitrary but should not contain special characters nor spaces. For example, the DB could be named **HeraldAckTriggers**, and inside the DB, one can define as many Bool data-points as one would like. For practicality reasons, we suggest that you name them **t1, t2, t3...** Use the **TriggerAckSet FB** (read 2.9.3.4.2.1) to connect your process actions (what will happen when the Trigger gets acknowledged) to this Data-block’s internal data-points.

One DB should be defined for all process variables. In this DB, for every process value you would like to include in your Herald notifications, you should define a data-point. The DB and data-point names are arbitrary but should not contain special characters nor spaces. For example, the DB could be named **HeraldVariables**, and inside the DB, one can define as many data-points as one would like. For practicality reasons, we suggest that you name them **v1, v2, v3...**

The arbitrarily given names must correspond to the...

...Trigger, Trigger acknowledgement and Variable names defined on the Herald Dashboard side in the ‘add new Trigger’ and ‘add new Variable’ dialogs!

## 2.9.3.4.2.3 Ethernet addresses setup

To configure your PLC to communication with Herald services, in the Device configuration section, open the Ethernet addresses subsection for the corresponding PLC. Define the IP address for your PLC and also enter the Subnet mask. Since your router will have to be configured to forward packages to your PLC, you will also have to enter the IP address of your router. Figure 2.27 shows an example of such a setup. In this case, the PLC's address is set to 192.168.0.171 while the Router's IP address is set to 192.168.0.1.

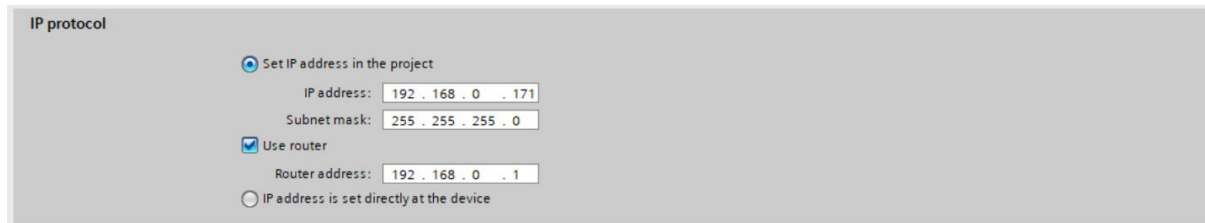


Figure 2.27 - TIA Portal Ethernet addresses setup

## 2.9.3.4.2.4 Web server setup

To use the PLC's internal web server, place a WWW block for synchronizing the user pages in your OB1. Usually, the control DB's number is 333 (read 2.9.3.4.2.5) so enter 333 or the corresponding value at the CTRL\_DB input. Figure 2.28 shows the OB1 Network 1 of our example application.

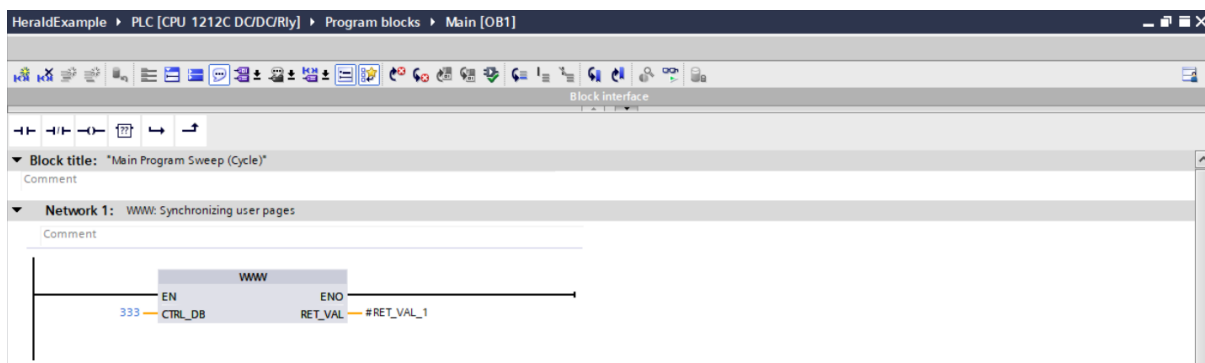


Figure 2.28 - TIA Portal WWW block

In the Device configuration section, Web server subsection, check the 'Activate Web server on all modules on this device' checkbox.

**Do not check the Permit access only with HTTPS checkbox!**

This option will permit communication only with those clients which install the PLC's internally available SSL certificate. This option is mainly intended for access through web browsers. Once Herald support's this option, we will send a notice to all of our users to update this setting if they wish to do so.

For Herald to be allowed to access the PLC's user defined web pages, the Integration's Username and Password must be defined. For this reason, a dedicated Herald user must be defined within the PLC's Web server's User management subsection.

Figure 2.29 shows an example of such a setup. We suggest to define Web server user Name with no special characters or spaces (for example 'Herald'). The access level should include only the **"open user-defined web pages"** and **"write in user-defined web pages"** options (evaluates to a restricted access level). Set the password to your preferences. The exact same Username and Password must be entered into the corresponding Integration definition fields.

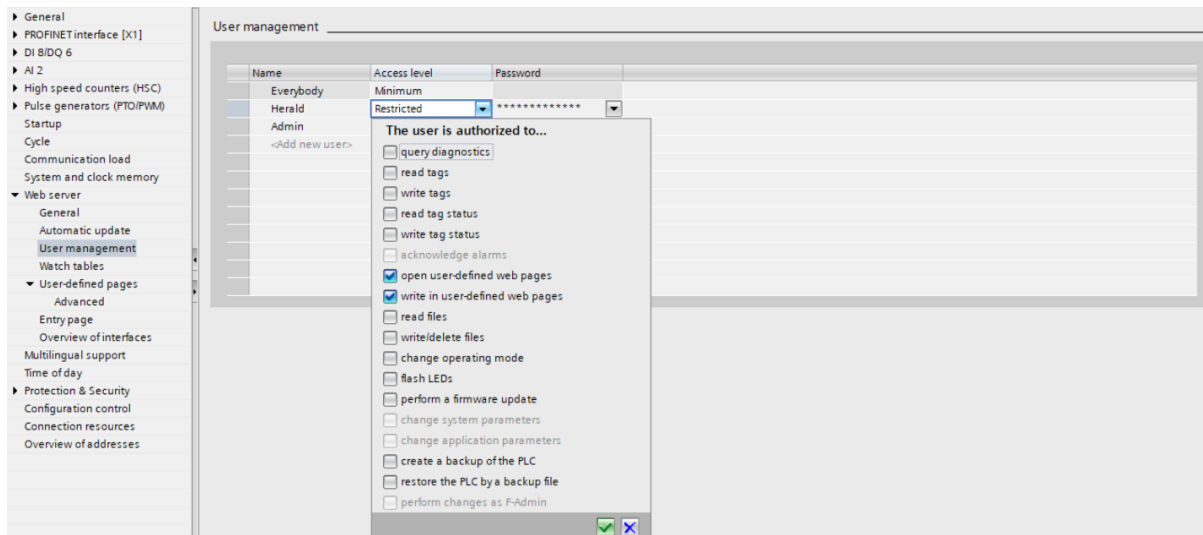


Figure 2.29 - TIA Portal User management setup

Since user defined web-pages can be accessed locally, it is very important **not to use the user dedicated for Herald Services for local access!** The PLC's web server can handle only a limited number of sessions which can be used up due to user's not logging out. For the Herald dedicated user, Herald services are taking care of logouts, so to prevent any issues with accessing the PLC remotely with this user...

**...do not use the user dedicated for Herald Services to directly access user defined web pages!**

When you are accessing the user defined web pages directly (via a browser) with a password protected account, make sure to...

**...log out as that user every time you finish work!**

Also, since the internal PLC's web server is handling the user authorization cookies and their validity using its internal time settings, it is important to...

**...keep the internal PLC time up to date!**

#### 2.9.3.4.2.5 User-defined pages setup

In the User-defined pages subsection, populate the 'HTML directory' entry with the exact path to the folder where you will place the .htm file you have downloaded from the Integrations overview in your Herald Dashboard (read 2.9.7). This can be any path, but it might be practical to create a 'User Pages' folder in the same place your TIA Portal project is located.

In the same subsection, populate the 'Default HTML page' entry with 'index.htm', or some other page to your preference.

Also, define an **Application name**. This entry must correspond to the orange part of the example URL which is explained in section 2.9.3.

Once the HTML directory, Default HTML page and the Application name have been defined, you can generate the User-defined pages DB blocks by clicking on the Generate blocks button. This will translate all of your setup to the corresponding DBs (usually DB333 and DB334). Once the blocks have been generated, you can download them to your PLC.





## 2.9.4 Add API (Server) Integration

By clicking the Add button in the Add/Refresh section, the Add new Integration dialog appears (Figure 2.30).

DEVICE TYPE	NAME	DESCRIPTION	ORGANIZATION	URL / UNIQUE KEY	SUBSCRIPTION	CREATED ON
Simatic S7-1x00	Devoted PLC	Test S7-1200 PLC	Devoted	https://herald-iot-simula...	Demo	30.09.2019

Figure 2.30 - Add new Integration dialog

DEVICE TYPE	NAME	DESCRIPTION	ORGANIZATION	URL / UNIQUE KEY	SUBSCRIPTION	CREATED ON	PING	ACT / DEACT	DOWNLOAD	EDIT	DELETE
Simatic S7-1x00	Devoted PLC	Test S7-1200 PLC	Devoted	https://herald-iot-simula...	Demo	30.09.2019 14:16					
Server	Devoted API	Test API	Devoted	*****	N/A	30.09.2019 14:29					

Figure 2.31 – Overview section with the added Direct Integration

The Integration type dropdown allows the User to select the Direct or API Integration type. Changes to this selector directly translate to changes in the Device Type dropdown (**Direct** to **Simatic S7-1x00**, **API** to **Server** and vice versa). Obviously, to add an API Integration, the **API** option needs to be selected.

The Save icon is disabled as long as values are not entered into mandatory fields. In this case, Integration name is the only mandatory field. The Organization selection dropdown is preset to one of the previously added Organizations or, as in this case, to the pinned Organization. As previously explained, if an Organization has been pinned, all Organization selection dropdown throughout the entire Dashboard will be preset to this Organization and disabled for editing. The Integration will be assigned to the Organization set in the Organization selection dropdown. Even though one can save an Integration with values entered only for mandatory fields, we strongly suggest that all values are entered since they are valuable for further setup and support.

Once mandatory fields have values, the Save button becomes enabled. By clicking on the Save button, the user Saves the Integration. Figure 2.31 shows the overview section with the added API Integration.

For every newly added API Integration, a unique key will be generated. This key will be used to make requests to the Herald API (read 2.9.4.2, the ioTUniqueKey parameter).

### 2.9.4.1 Subscription handling

All newly added API Integrations are automatically assigned to the preexisting API subscription. By default, every new Organization gets a **100% free Demo API Subscription** which can be used for testing purposes. Unlike Direct Integrations, multiple API Integrations can be assigned to one API Subscription. Under the Demo API Subscription, 100 free API requests can be sent. After this, the subscription will become usage deactivated.



The Demo API Subscription can be upgraded to the **Classic API Subscription**. In that case, all API Integrations which have been assigned to the Demo Subscription will automatically get re-assigned to the upgraded Classic subscription. Under the Classic API subscription, all assigned Integration can work for as long as there are enough resources in the Organization's account. The pricing for the Classic API subscription can be viewed in the [Pricing Policy Documentation](#).

#### 2.9.4.2 Configure Client Server

To make requests to the created API Integration, Users must configure their services to make requests according to the Herald API documentation. Once configured, the User's services will be able to signal changes to defined Triggers and send Variables values which will be used to generate notifications for the client's Users.

In the following table, a request specification is given for triggering an added Trigger, assigned to an API Integration.

Description	Request triggering of a configured Trigger, assigned to an API Integration	
Protocol	HTTPS	
URI	https://prod-herald-public-api.azurewebsites.net/v1/events	
Method	POST	
Headers	KEY: Content-Type	VALUE: application/json
	KEY: Authorization	VALUE: Basic YXBpOjA4QUREODQwMEFFNTQ2OEFCRUl0QzAyQjc3MUYyQjZl
Body*	<pre>{   "ioTUniqueKey": "xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx",   "triggerUniqueKey": "xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx ",   "triggerValue": 1,   "variables": [{     "variableName": "xxxxxxxx", "variableValue": "xxxxxxxx"   }] }</pre>	

The previous request's Body is a JSON object populated with data which can be found in the Dashboard.

Property	Type	Description
ioTUniqueKey	string	Unique Integration identification key, auto-generated for all added API Integrations. <b>The key can be copied from the Integrations Overview page.</b> Only administrator users have access to it. This identifier should be kept a secret! Based on this parameter, processed events will be logged and all consequential costs assigned to the corresponding Organization. To change the parameter, please contact support@devoted.ba.
triggerUniqueKey	string	Unique Trigger identification, auto-generated for all added API Integration Triggers. <b>The key can be copied from the Integration Triggers Overview page.</b>
triggerValue	int	Optional parameter with the default value set to 1. It can be set to any value that is of use to your organization.
Variables	Array of Variable objects (name-value pairs)	Array of all configured Variables under the corresponding API Integration. Each variable is an Object with two properties, variableName (as defined in Herald Dashboard) and variableValue (the current value of the defined variable).



The Herald API responds to requests with following status codes:

Status	Message	Explanation
400	Bad Request	Missing request object or request object could not be deserialized due to incorrect format. Request object not compliant with data annotation rules.
404	Not Found	No Trigger found for specified trigger unique key
405	Method Not Allowed	Current subscription status or Organization balance do not allow processing of request. Check the status of the Organization's API subscription and the balance of the Organization's Herald account.
500	Internal Server Error	Handled or unhandled exception occurred not related to the incoming request or user or organization specific configuration. Contact support@devoted.ba to get more information and resolve problem if possible.

#### 2.9.4.2.1 Configure WinCC Server

All WinCC servers with a connection to the internet can be integrated with the Herald Notification Platform as an API Integration. To do so, the user must leverage the scripting functionality available within the WinCC environment. Scripts can be configured to run:

- locally - on a given page, triggered to run as a consequence of an event (a Tag value change let say) or
- globally - independent of the current page, triggered to run cyclically by a configured timer or just as a consequence of an event (a Tag value change let say).

In both cases, a simple VBS script can be used to send API requests to the Herald Platform API. The following example shows an API request which will be sent to the Herald Platform API every time a Tag with the name 'Test\_Tag' changes its value to other than 0.

```
Option Explicit
Function action

Dim TagObject
Dim Value
Set TagObject = HMIRuntime.Tags("Test_Tag")
Value = TagObject.Read

If Value <> 0 Then

TagObject.Write 0

Dim objXMLHTTP
Set objXMLHTTP = CreateObject("MSXML2.ServerXMLHTTP.6.0")

Dim params
params = "{ 'IoTUniqueKey': 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx', 'triggerUniqueKey': 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx', 'triggerValue': 1, 'variables': [{ 'variableName': 'Your_Variable_Name', 'variableValue': '178' }] }"

objXMLHTTP.open "POST", "http://prod-herald-public-api.azurewebsites.net/v1/events", "false"
objXMLHTTP.setRequestHeader "Authorization", "Basic YXBpOjA4QUREODQwMEFFNTQ2OEFRCUI0QzAyQjc3MUYYQjZF"
objXMLHTTP.setRequestHeader "Content-Type", "application/json"
objXMLHTTP.send params

End If

End Function
```

In this case, the script is configured to run globally, triggered cyclically by a timer, every 10 seconds. Figure 2.32 shows the timer configuration dialog in WinCC V7.2.

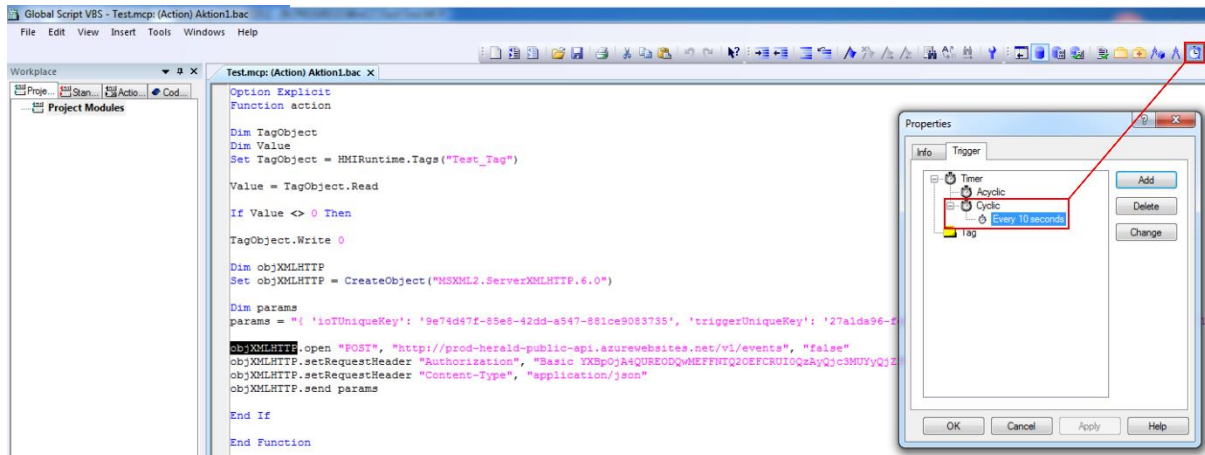


Figure 2.32 - WinCC 7.2 global VBS script trigger setup

Basically, the script checks the current value of the tag with name "Test\_Tag" every 10 seconds. If the value is not 0, the tag value is set to 0, an API request is created and sent to the Herald Services API. The **params** variable and the properties of the **objXMLHTTP** object are set according to the general instructions given in 2.9.4.2.

#### 2.9.4.2.2 Configure WinCC Flexible Server

All WinCC Flexible servers with a connection to the internet can be integrated with the Herald Notification Platform as an API Integration. To do so, the user must leverage the scripting functionality available within the WinCC Flexible environment. Scripts can be configured to run:

- locally - on a given page, triggered to run as a consequence of an event (a Tag value change let say) or
- globally - independent of the current page, triggered to run as a consequence of an event (a Tag value change let say).

In both cases, a simple VBS script can be used to send API requests to the Herald Platform API. The following example shows an API request which will be sent to the Herald Platform API every time a Tag with the name 'Test\_Tag' changes its value to other than 0.

```

If HmiRuntime.SmartTags("Test\Test_Tag") <> 0 Then

HmiRuntime.SmartTags("Test\Test_Tag") = 0

Dim objXMLHTTP
Set objXMLHTTP = CreateObject("MSXML2.ServerXMLHTTP.6.0")

Dim params
params = "{ 'ioUniqueKey': '9e74d47f-85e8-42dd-a547-881ce9083735', 'triggerUniqueKey': '27a1da96-fc6a-4c19-905f-9a06c9587eb4', 'triggerValue': 1, 'variables': [{ 'variableName': 'Random character', 'variableValue': '789' } ] }"

objXMLHTTP.open "POST", "http://prod-herald-public-api.azurewebsites.net/v1/events", "false"
objXMLHTTP.setRequestHeader "Authorization", "Basic YXBpOjA4QURERODQwMEFFNTQ2OEFRCUI0QzAyQjc3MUYYQjZF"
objXMLHTTP.setRequestHeader "Content-Type", "application/json"
objXMLHTTP.send params

End If

```

In this case, the script is configured to run globally, triggered every time the Test\_Tag changes value. This can easily be configured from the Tag overview screen.

Figure 2.33 shows the tag-script connection configuration dialog in WinCC Flexible 2008 SP3 Upd5.

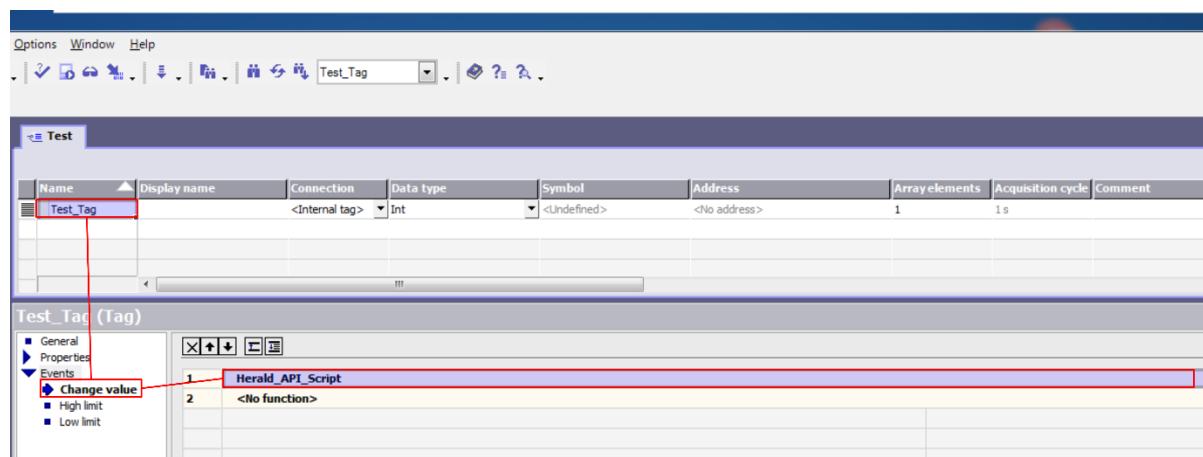


Figure 2.33 - WinCC Flexible VBS script trigger setup

Basically, the script checks the current value of the tag with name “Test\_Tag” every time the tag value changes. If the value is not 0, the tag value is set to 0, an API request is created and sent to the Herald Services API. The **params** variable and the properties of the **objXMLHTTP** object are set according to the general instructions given in 2.9.4.2.

## 2.9.5 Ping Integration

Only Direct (Simatic S7-1x00) Integrations can be pinged since for their functioning, an URL entry is required. This option is helpful for testing the network setup when establishing communication from Herald services to your PLC.

By clicking on the button in the Ping column, Herald services will try to initiate communication to the defined URL. If a response is received, the entry in the corresponding table row will become a green check icon. Otherwise, it will become a red cross, indicating that no response from the entered URL has been received. In this case, the corresponding Direct integration will not operate.

## 2.9.6 Activate/Deactivate Integration

Only Direct (Simatic S7-1x00) Integrations can be Activated/Deactivated since only the communication for this type of integrations is initiated, established and maintained from the Herald services. For a Direct Integration to be Activated/Deactivated, it must be assigned to a subscription (Demo or Classic).

The button in the Act/Deact column indicates the current state of the Integration. If the button is gray, the Integration is currently deactivated. If the button is green, the Integration is currently activated.

By clicking on the button in the Act/Deact column, the Activate/Deactivate dialog will appear for the corresponding Integration. If the user has insufficient permissions to Activate/Deactivate an Integration, the button for the corresponding Integration will be disabled. Only administrator users can Activate/Deactivate Integrations.

Since the Activation/Deactivation of an Integration can have impact on the functioning of the entire notification system’s setup and also on the billed amount, the User is required to enter the exact name of the Integration he/she would like to Activate/Deactivate. Once the exact name of the Integration is entered, the Activate/Deactivate button becomes enabled. By clicking on the button, the User Activates or Deactivates the corresponding Integration.



## 2.9.7 Download Integration

For Direct (Simatic S7-1x00) Integrations, the download option will download an .htm file, **heraldnotifications.htm**.

When the download button is clicked, all Herald Dashboard settings related to this Integration (Integration specific data, linked Triggers and Variables) are evaluated and included in a .htm file which needs to be placed in the directory which is selected as the HTML directory in the User-defined pages setup section of your TIA Portal project (read 2.9.3.4.2.5).

After the file is placed, the PLC's web server Data-blocks need to be regenerated and downloaded to the device. This process is called deploying the file to the PLC's web server.

Once the web server Data-blocks are generated, downloaded and the PLC started, this file will become a page under the PLC's web server which Herald services will get and evaluate once every minute (using the provided credentials), provided the Integration is activated. If you analyze the previously mentioned URL parameter which needs to be defined for Direct Integrations (see section 2.9.3), you will realize that the required URL format is defined to point to this file on your PLC's web server.

Since this file is the only way that Herald services can communicate with your PLC, it is important that you...

**...do not manually edit the file's content,**

**...download and redeploy the file to your PLC after every Integration related change on the Dashboard.**

All changes to the Integration itself, Variables and Triggers under that Integration are considered Integration related changes. Technically, changes to some values like descriptions do not require a redeployment since they are not even included in the file, but...

**...adding, deleting or changing of addresses must be followed by an Integration download, and PLC redeployment!**

## 2.9.8 Edit Integration

By clicking on the button in the Edit column, the corresponding Integration will be loaded into the Edit dialog for editing. If the user has insufficient permissions to edit an Integration, the edit button for the corresponding Integration will be disabled. Only administrator users can edit Integrations.

DEVICE TYPE	NAME	DESCRIPTION	ORGANIZATION	URL / UNIQUE KEY	SUBSCRIPTION	CREATED ON	PING	ACT / DEACT	DOWNLOAD	EDIT	DELETE
Simatic S7-1x00	Devoted PLC	Test S7-1200 PLC	Devoted	https://herald-iot-simula...	Demo	30.09.2019 14:16	X	⊕	⬇	✎	✖
Server	Devoted API	Test API	Devoted	*****	N/A	30.09.2019 14:29	N/A	N/A	N/A	✎	✖

Figure 2.34 - Edit Integration dialog

Figure 2.34 shows the edit Integration dialog for the example Integration "Devoted PLC" within the "Devoted" Organization. Mandatory fields cannot be edited to be empty. Integration type, Device type and Organization dropdowns are disabled in the edit dialog.



To make changes to these fields, the Users needs to delete the Integration and add a new one with the wanted changes entered to the corresponding fields. The Save button becomes enabled once a change to any of the fields' values is detected. At any time, the editing can be canceled by clicking on the Cancel button.

## 2.9.9 Delete Integration

By clicking on the button in the Delete column, Herald services will initiate the deletion of the corresponding Integration. If the user has insufficient permissions to delete an Integration, the delete button for the corresponding Integration will be disabled. Only administrator users can delete Integrations.

Figure 2.14 shows that first step of the Integration deletion process.

**Herald Dashboard**

Dashboard / Integrations

Delete Cancel

The delete action is **irreversible**. For that reason we would like you to enter the exact name of the integration you would like to delete. Since integrations can have dependencies such as variables and triggers, we will check what dependencies exist for the integration you are trying to delete. If no dependencies are found, the **integration will be deleted**. If any dependencies are found, you will be asked to confirm the delete action for the integration and all of its dependencies by clicking on the 'Confirm delete' button. This will **delete the chosen integration** and **delete / unassign all the listed dependencies**. You can abort the delete process by clicking on the 'Cancel' button.

Enter the exact integration's name:

Enter name..

Integrations Time Triggers Triggers Variables

+ Add Refresh

Organizations: Devoted Search by name: Entry name..

DEVICE TYPE	NAME	DESCRIPTION	ORGANIZATION	URL / UNIQUE KEY	SUBSCRIPTION	CREATED ON
* Simatic S7-1x00	Devoted PLC	Test S7-1200 PLC	Devoted	https://herald-iot-simula...	Demo	30.09.2019
* Server	Devoted API	Test API	Devoted	*****	N/A	30.09.2019

Figure 2.35 - Delete Integration dialog

**The delete action is irreversible!** For that reason the user is required to enter the exact name of the Integration he/she would like to delete. Once the exact name of the Integration is entered, the Delete button becomes enabled. By clicking on the delete button, the user initiates the deletion process. Since Integrations can have dependencies such variables and triggers, the system will check what dependencies exist for the Integration the user is trying to delete. If no dependencies are found, the Integration will be deleted right away. If any dependencies are found, the user will be asked to confirm the delete action for the Integration and all of its dependencies by clicking on the 'Confirm delete' button. This will delete the chosen Integration and delete all the listed dependencies. The delete action can be aborted by clicking on the 'Cancel' button.

## 2.10 Time-triggers

### 2.10.1 About

As already explained in 1.1.6, Time-triggers are not linked to an Integration and as one would imagine, they are triggered by time. They can be used to trigger sending of daily or weekly notifications to remind users of time dependent events such as scheduled repetitive tasks, changes in shifts etc.

Time dependent events can be defined as:

- **Single occurrence type** events which will occur at some point in time in the future or
- **Recurring occurrence type** events which means that they will be triggered and then rescheduled over and over again for the future, according to the additional settings you make.

Recurring events can have an Exact or Relative occurrence time definition:

- For the **Exact occurrence time definition** of a recurring Time-trigger, the user must provide the Month, Day, Hour and Minute settings so that Herald services can calculate when the trigger will be triggered.
- For the **Relative occurrence time definition** of a recurring Time-trigger, the user must choose the starting point in time and define a delay time. When the delay time has passed, the trigger will execute for the first time and reschedule itself for future executions. The pauses between trigger executions will be equal to the defined delay time.

All pages accessed from the left navigation pane (read 2.4, numbered item 3) except the Subscriptions page are organized in three sections:

1. Add / Refresh section,
2. Filter section,
3. Overview section.

Figure 2.36 shows the Time-triggers overview page with the previously described sections. As you can see, the Time-triggers view has an additional section (used to navigate between different Integration-related configuration screens):

4. Integrations navigation section.

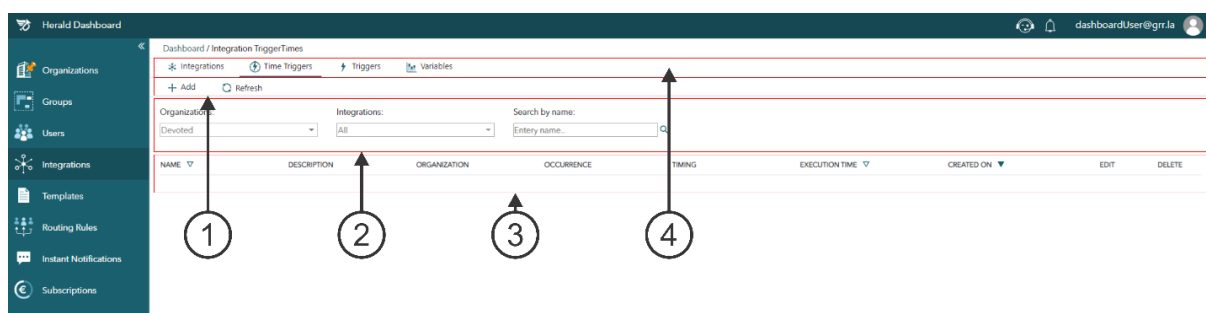


Figure 2.36 - Integrations Overview

By default, all currently defined Time-triggers will appear in the Overview section. Time-triggers can be filtered by editing the Filter section. By clicking the Refresh button in the Add/Refresh section, the Overview section is reloaded with the currently set filter values. Time-triggers can be filtered by their assignment to an Organization. Filtered Time-triggers can be searched by their name. These options can be found in the Filter section.

In the overview section, Time-triggers are displayed in a table where each Time-trigger is represented by one table row. Most of the table columns are dedicated for presenting the Time-trigger specific data (entered by the Time-trigger creator / editor). An interesting column to analyze is the execution time column which shows the next calculated execution time for the corresponding Time-trigger. The last two columns on the right side of the table are so called action columns. By clicking on the icon in the column, the user initiates an action for the Time-trigger in the corresponding table row. This way, Time-triggers can be Edited or Deleted.



## 2.10.2 Add new Time-trigger

By clicking the Add button in the Add/Refresh section, the Add new Time-trigger dialog appears (Figure 2.37).

Figure 2.37 - Add new Time-trigger dialog

The Save icon is disabled as long as values are not entered into mandatory fields. In this case, Time-trigger name is the only mandatory field. If the user would change the Occurrence type or time definition dropdown values, other fields might become mandatory as well but that will be self-explanatory.

The Organization selection dropdown is preset to one of the previously added Organizations or, as in this case, to the pinned Organization. As previously explained, if an Organization has been pinned, all Organization selection dropdown throughout the entire Dashboard will be preset to this Organization and disabled for editing. The Time-trigger will be assigned to the Organization set in the Organization selection dropdown.

Even though one can save a Time-trigger with values entered only for mandatory fields, we strongly suggest that all values are entered since they are valuable for further setup and support. Once mandatory fields have values, the Save button becomes enabled. By clicking on the Save button, the user saves the Time-trigger.

Figure 2.38 shows the overview section with one created Time-trigger.

NAME	DESCRIPTION	ORGANIZATION	OCCURRENCE	TIMING	EXECUTION TIME	CREATED ON	EDIT	DELETE
Single Time Trigger	Just a test	Devoted	Single	Exact	10.10.2019 17:47	10.10.2019 17:32		

Figure 2.38 – Overview section with one created Time-trigger

### 2.10.2.1 Subscription handling

Depending on the Occurrence type and Time definition settings, every added Time-trigger will create at least one scheduled event (a moment in time in the future when the configured Time-trigger will get triggered). Every triggered scheduled event will either be billed or subtracted from the number of free scheduled events, depending on the currently active subscription.



By default, every new Organization gets a **100% free Demo Scheduled Event Subscription** which can be used for testing purposes. Under the Demo Scheduled Event Subscription, 5 free Scheduled Events can be added. After the events get triggered, the subscription will become usage deactivated.

The Demo Scheduled Event Subscription can be upgraded to the **Consumption Scheduled Event Subscription**. Under the Consumption Scheduled Event subscription, an unlimited number of Scheduled events can be defined, but events will only get triggered for as long as there are enough resources in the Organization's account. The pricing for the Consumption Scheduled Event Subscription can be viewed in the [Pricing Policy Documentation](#).

### 2.10.3 Edit Time-trigger

By clicking on the button in the Edit column, the corresponding Time-trigger will be loaded into the Edit dialog for editing. If the user has insufficient permissions to edit a Time-trigger, the edit button for the corresponding Time-trigger will be disabled. Only administrator users can edit Time-triggers.

Figure 2.39 shows the edit Time-trigger dialog for the example "Single Time trigger" within the "Devoted" Organization. Mandatory fields cannot be edited to be empty. The Save button becomes enabled once a change to any of the fields' values is detected. At any time, the editing can be canceled by clicking on the Cancel button. The Time-trigger's assignment to an Organization cannot be changed (to do this, delete the first Time-trigger, and add another one to the other Organization).

The screenshot shows the 'Edit Time-trigger' dialog in the Herald Dashboard. The dialog has a sidebar with navigation links: Organizations, Groups, Users, Integrations, Templates, Routing Rules, Instant Notifications, and Subscriptions. The main content area is titled 'Dashboard / Integration TriggerTimes' and has tabs for Integrations, Time Triggers (selected), Triggers, and Variables. The dialog contains the following fields:

- Name (unique within integration): Single Time Trigger
- Description: Just a test
- Organization: Devoted
- Occurrence type: Single
- Set event date-time: 10.10.2019 17:47

Below the fields is a table of existing time triggers. An arrow points to the 'EDIT' button in the table.

NAME	DESCRIPTION	ORGANIZATION	OCCURRENCE	TIMING	EXECUTION TIME	CREATED ON	EDIT	DELETE
Single Time Trigger	Just a test	Devoted	Single	Exact	10.10.2019 17:47	10.10.2019 17:32	EDIT	DELETE

Figure 2.39 - Edit Time-trigger dialog

### 2.10.4 Delete Time-trigger

By clicking on the button in the Delete column, Herald services will initiate the deletion of the corresponding Time-trigger. If the user has insufficient permissions to delete a Time-trigger, the delete button for the corresponding Time-trigger will be disabled. Only administrator users can delete Time-triggers.

**The delete action is irreversible!** For that reason the user is required confirm the request by clicking on the 'Delete' button. By clicking on the delete button, the user initiates the deletion process. Since Time-triggers can have dependencies such as assignment to routing rules, the system will check what dependencies exist for the Time-trigger the user is trying to delete. If no dependencies are found, the Time-trigger will be deleted right away. If any dependencies are found, the user will be asked to confirm the delete action for the Time-trigger and all of its dependencies by clicking on the 'Confirm delete' button. This will delete the chosen Time-trigger and unassign all listed dependencies. The delete action can be aborted by clicking on the 'Cancel' button.

## 2.11 Triggers

All pages accessed from the left navigation pane (read 2.4, numbered item 3) except the Subscriptions page are organized in three sections:

1. Add / Refresh section,
2. Filter section,
3. Overview section.

Figure 2.40 shows the Triggers overview page with the previously described sections. As you can see, the Triggers view has an additional section:

4. Integrations navigation section.

As previously explained, this section is used to navigate between different Integration-related configuration screens.

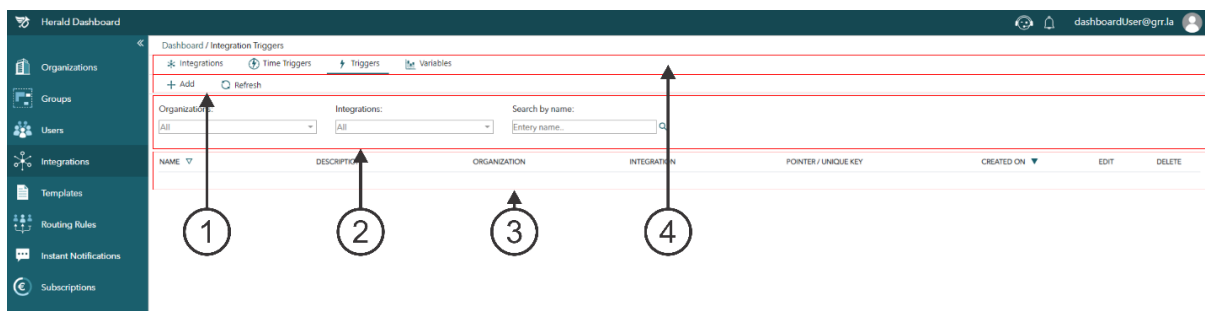


Figure 2.40 - Integrations Overview

By default, all currently defined Triggers will appear in the Overview section. Triggers can be filtered by editing the Filter section. By clicking the Refresh button in the Add/Refresh section, the Overview section is reloaded with the currently set filter values. Triggers can be filtered by their assignment to Organizations and Integrations. Filtered Triggers can be searched by their name. These options can be found in the Filter section.

In the overview section, Triggers are displayed in a table where each Trigger is represented by one table row. Most of the table columns are dedicated for presenting the Trigger specific data (entered by the Trigger creator / editor). The last two columns on the right side of the table are so called action columns. By clicking on the icon in the column, the user initiates an action for the Trigger in the corresponding table row. This way, Triggers can be Edited or Deleted.

### 2.11.1 Add new Trigger

#### 2.11.1.1 About

In the Herald Dashboard environment, sending of notifications is initiated by modifying the values of special variables which are called **Triggers**. In this document, the modification of these Trigger variables will often be referred to as triggering the Trigger. While Time-triggers (read 1.1.6 and 2.10) are linked to time and have really no connection with Integrations, other Triggers are closely coupled with Integrations. Since Herald services support two types of Integrations (read 2.9), two types of Triggers can be defined:

1. **Direct Integration Triggers.** These triggers are linked to a previously defined Direct Integration. They are defined as links to values of Data-block (DB) data-points within the PLC. If a DB data-point change is detected, the corresponding Trigger is being triggered.
2. **API Integration Triggers.** These triggers are linked to a previously defined API integration. They are triggered from the client side via https requests.

## 2.11.1.2 Add new Direct Integration Trigger

By clicking the Add button in the Add/Refresh section, the Add new Trigger dialog appears. If the Integration dropdown is set to a previously defined Direct Integration, the UI will offer S7-1x00 specific setup as shown in Figure 2.41.

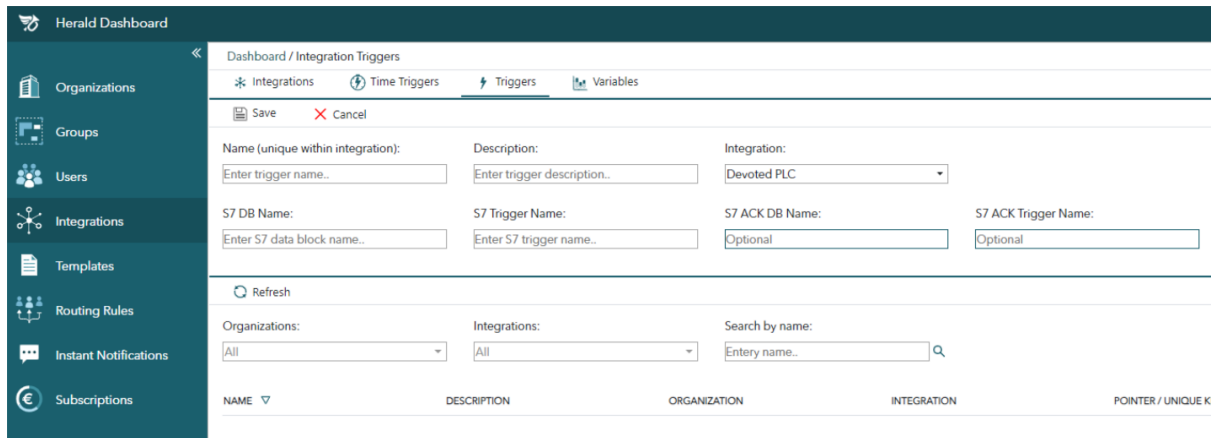
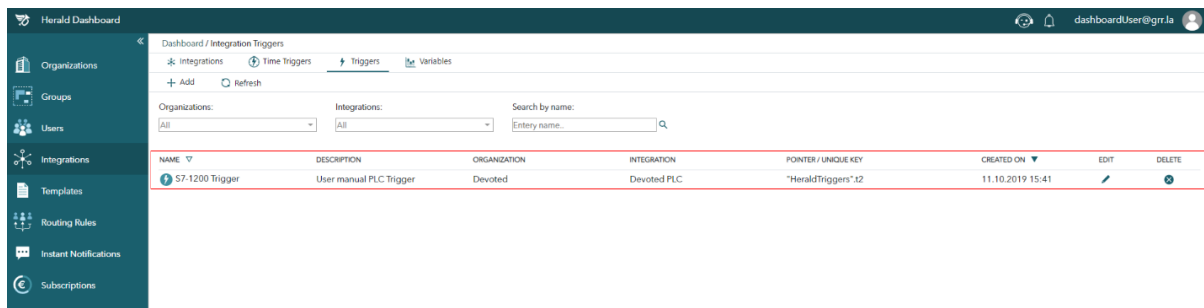


Figure 2.41 - Add new Trigger dialog



NAME	DESCRIPTION	ORGANIZATION	INTEGRATION	POINTER / UNIQUE KEY	CREATED ON	EDIT	DELETE
S7-1200 Trigger	User manual PLC Trigger	Devoted	Devoted PLC	"HeraldTriggers".t2	11.10.2019 15:41		

Figure 2.42 – Overview section with one created Trigger

The Save icon is disabled as long as values are not entered into mandatory fields. In this case, **Trigger name**, **S7 DB Name** and **S7 Trigger Name** are mandatory fields. Even though one can save a Trigger with values entered only for mandatory fields, we strongly suggest that all values are entered since they are valuable for further setup and support. Once mandatory fields have values, the Save button becomes enabled. By clicking on the Save button, the user saves the Trigger. Figure 2.42 shows the overview section with one created Trigger.

Regarding the **S7 DB Name** and **S7 Trigger Name** fields, they are to be populated according to the corresponding Data-block name and data-point name in your TIA Portal application. For example (as described in 2.9.3.4.2.2), if the corresponding DB in your TIA Portal application is named **HeraldTriggers**, then the **S7 DB Name** should be populated with the exact same value, that is, **HeraldTriggers**. Also, if the data-point within that DB is named, let's say **t2**, the **S7 Trigger Name** field should be populated with that exact same value, that is, **t2**.

As described in 2.9.3.4.2.1 and 2.9.3.4.2.2, triggers can be acknowledged via the received corresponding Herald app notifications. Sometimes, it is useful to configure an action to be initiated on the PLC when a notification gets acknowledged. For this purpose, for each Direct Integration Trigger the User can optionally define an acknowledgement DB and data-point, **S7 ACK DB Name** and **S7 ACK Trigger Name**. These fields are populated similarly to the previous example. So (as described in 2.9.3.4.2.2), if the corresponding DB in your TIA Portal application is named **HeraldAckTriggers**, then the **S7 ACK DB Name** should be populated with the exact same value, that is, **HeraldAckTriggers**. Also, if the data-point within that DB is named, let's say **t2**, the **S7 ACK Trigger Name** field should be populated with that exact same value, that is, **t2**.

For a Trigger defined as previously described, every time a change would be detected in the value of the "HeraldTriggers".t2 data-point on the PLC side, this Trigger would be triggered on the Herald services side. Supposed that Templates and Routing Rules had previously been defined so that some user would receive a



notification due to this Trigger being triggered. If the user would acknowledge this notification by clicking on the acknowledge button, the “HeraldAckTriggers”.t2 data-point on the PLC side would be set to value “True”.

### 2.11.1.3 Add new API Integration Trigger

By clicking the Add button in the Add/Refresh section, the Add new Trigger dialog appears. If the Integration dropdown is set to a previously defined API Integration, the UI will offer API specific setup as shown in Figure 2.43.

Figure 2.43 - Add new Trigger dialog

The Save icon is disabled as long as values are not entered into mandatory fields. In this case, Trigger name is the only mandatory field. Even though one can save a Trigger with values entered only for mandatory fields, we strongly suggest that all values are entered since they are valuable for further setup and support. Once mandatory fields have values, the Save button becomes enabled. By clicking on the Save button, the user saves the Trigger.

Regarding the **Trigger Ack. Webhook** field, this is an optional field which is used for the acknowledgement functionality for the API Integration originated notifications. If left empty, the acknowledgement of a corresponding notification will have no consequences. If a Webhook is defined and a notification which has been sent as a consequence of this Trigger being triggered is acknowledged, all available event specific data will be posted to this Webhook in JSON format. Obviously, if populated, the Webhook must be a valid URL.

To test your Webhooks, you can [visit third party sites](#) which will allow you to test more easily and configure your real system according to your needs.

Figure 2.44 shows the overview section with one Direct Integration and one API Integration Trigger. In the “POINTER/UNIQUE KEY” column, for every API trigger, a unique key value will be available. A click on the field will copy the value to clipboard. To trigger the corresponding Trigger, this KEY needs to be defined in the API request (read 2.9.4.2). The value of the KEY can be set to 1.

NAME	DESCRIPTION	ORGANIZATION	INTEGRATION	POINTER / UNIQUE KEY	CREATED ON	EDIT	DELETE
API Trigger	User manual API Trigger	Devoted	Devoted API	eacd7ad0-b315-4a82-a062-992d3c6f4...	11.10.2019 16:03		
S7-1200 Trigger	User manual PLC Trigger	Devoted	Devoted PLC	"HeraldTriggers".t2	11.10.2019 15:41		

Figure 2.44 – Overview section with one created Trigger



## 2.11.2 Edit Trigger

By clicking on the button in the Edit column, the corresponding Trigger will be loaded into the Edit dialog for editing. If the user has insufficient permissions to edit a Trigger, the edit button for the corresponding Trigger will be disabled. Only administrator users can edit Triggers.

Figure 2.45 shows the edit Trigger dialog for the example Trigger “API Trigger” for the “Devoted API” Integration. Mandatory fields cannot be edited to be empty. The Save button becomes enabled once a change to any of the fields’ values is detected. At any time, the editing can be canceled by clicking on the Cancel button. The Trigger’s assignment to an Integration cannot be changed (to do this, delete the first Trigger, and add another one to the other Integration).

The screenshot shows the 'Edit Trigger' dialog in the Herald Dashboard. The dialog has a sidebar with navigation links: Organizations, Groups, Users, Integrations, Templates, Routing Rules, Instant Notifications, and Subscriptions. The main area is titled 'Dashboard / Integration Triggers' and has tabs for Integrations, Time Triggers, Triggers, and Variables. The 'Triggers' tab is active. At the top of the dialog are 'Save' and 'Cancel' buttons. Below these are input fields for 'Name (unique within integration):', 'Description:', and 'Integration:'. The 'Name' field contains 'API Trigger', 'Description' contains 'User manual API Trigger', and 'Integration' is a dropdown menu set to 'Devoted API'. Below these are fields for 'Trigger unique key (auto-generated):' and 'Trigger Ack. Webhook:'. The 'Trigger unique key' field contains a long alphanumeric string, and the 'Trigger Ack. Webhook' field contains a URL. Below these fields is a 'Refresh' button. At the bottom of the dialog is a table with columns: NAME, DESCRIPTION, ORGANIZATION, INTEGRATION, POINTER / UNIQUE KEY, CREATED ON, and EDIT. The table contains two rows: 'API Trigger' and '57-1200 Trigger'. The 'API Trigger' row is highlighted. An arrow points to the 'EDIT' button in the 'API Trigger' row.

NAME	DESCRIPTION	ORGANIZATION	INTEGRATION	POINTER / UNIQUE KEY	CREATED ON	EDIT	DELETE
API Trigger	User manual API Trigger	Devoted	Devoted API	eaed7ad0-b315-4a82-a062-992d3c664...	11.10.2019 16:03		
57-1200 Trigger	User manual PLC Trigger	Devoted	Devoted PLC	"HeraldTriggers":t2	11.10.2019 15:41		

Figure 2.45 - Edit Trigger dialog

## 2.11.3 Delete Trigger

By clicking on the button in the Delete column, Herald services will initiate the deletion of the corresponding Trigger. If the user has insufficient permissions to delete a Trigger, the delete button for the corresponding Trigger will be disabled. Only administrator users can delete Triggers.

**The delete action is irreversible!** For that reason the user is required confirm the request by clicking on the 'Delete' button. By clicking on the delete button, the user initiates the deletion process. Since Triggers can have dependencies such as assignment to routing rules, the system will check what dependencies exist for the Trigger the user is trying to delete. If no dependencies are found, the Trigger will be deleted right away. If any dependencies are found, the user will be asked to confirm the delete action for the Trigger and all of its dependencies by clicking on the 'Confirm delete' button. This will delete the chosen Trigger and unassign all listed dependencies. The delete action can be aborted by clicking on the 'Cancel' button.

## 2.12 Variables

All pages accessed from the left navigation pane (read 2.4, numbered item 3) except the Subscriptions page are organized in three sections:

1. Add / Refresh section,
2. Filter section,
3. Overview section.

Figure 2.46 shows the Variables overview page with the previously described sections. As you can see, the Variables view has an additional section:

4. Integrations navigation section.

As previously explained, this section is used to navigate between different Integration-related configuration screens.

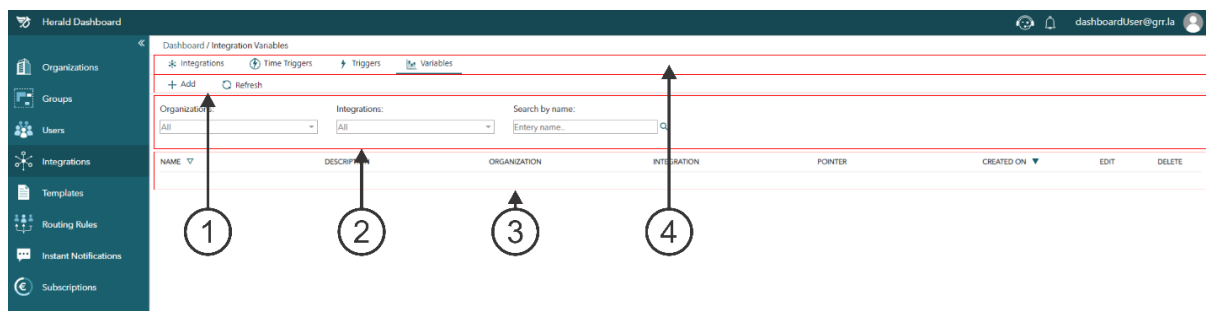


Figure 2.46 - Integrations Overview

By default, all currently defined Variables will appear in the Overview section. Variables can be filtered by editing the Filter section. By clicking the Refresh button in the Add/Refresh section, the Overview section is reloaded with the currently set filter values. Variables can be filtered by their assignment to Organizations and Integrations. Filtered Variables can be searched by their name. These options can be found in the Filter section.

In the overview section, Variables are displayed in a table where each Variable is represented by one table row. Most of the table columns are dedicated for presenting the Variable specific data (entered by the Variable creator / editor). The last two columns on the right side of the table are so called action columns. By clicking on the icon in the column, the user initiates an action for the Variable in the corresponding table row. This way, Variables can be Edited or Deleted.

### 2.12.1 Add new Variable

#### 2.12.1.1 About

In the Herald Dashboard environment, sending of notifications is initiated by triggering of **Triggers**. Now, when a Trigger is triggered, it is often useful to include additional values in Notifications so that the delivered message is as informative as possible (process or business related data). This can be done using Variables. Since Herald services support two types of Integrations (read 2.9), two types of Variables can be defined:

1. **Direct Integration Variables.** These variables are linked to a previously defined Direct Integration. They are defined as links to values of Data-block data-points within the PLC. Whenever a Trigger change on the PLC is detected, all Variables are read so that they can be included in notifications.
2. **API Integration Variables.** These variables are linked to a previously defined API integration. They are defined and sent to the platform as an integral part of the notification triggering request.

## 2.12.1.2 Add new Direct Integration Variable

By clicking the Add button in the Add/Refresh section, the Add new Variable dialog appears. If the Integration dropdown is set to a previously defined Direct Integration, the UI will offer S7-1x00 specific setup as shown in Figure 2.47.

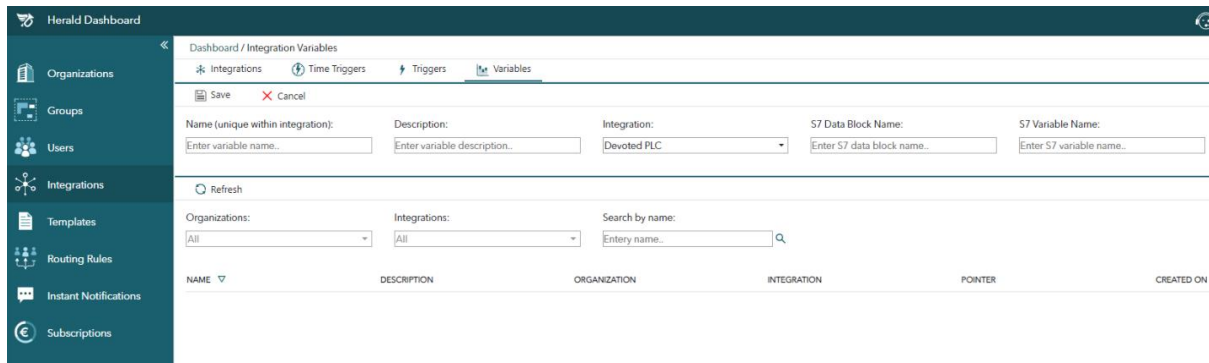
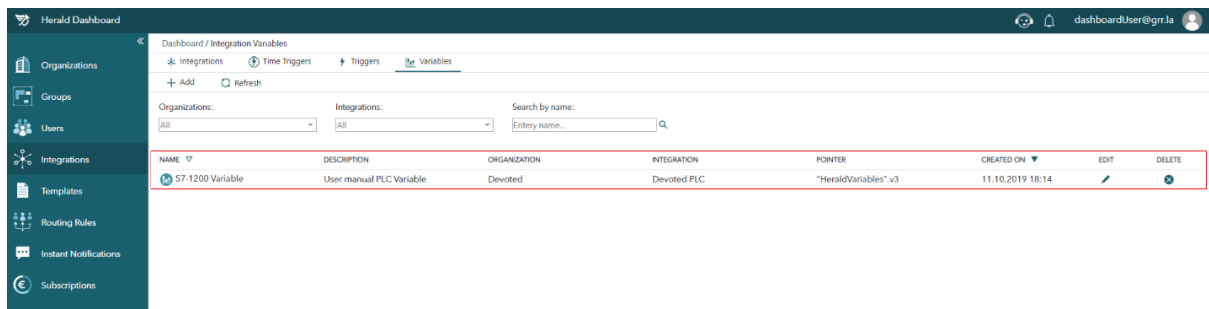


Figure 2.47 - Add new Variable dialog

The Save icon is disabled as long as values are not entered into mandatory fields. In this case, **Variable name**, **S7 DB Name** and **S7 Variable Name** are mandatory fields. Even though one can save a Variable with values entered only for mandatory fields, we strongly suggest that all values are entered since they are valuable for further setup and support. Once mandatory fields have values, the Save button becomes enabled. By clicking on the Save button, the user saves the Variable.

Regarding the **S7 DB Name** and **S7 Variable Name** fields, they are to be populated according to the corresponding Data-block name and data-point name in your TIA Portal application. For example (as described in 2.9.3.4.2.2), if the corresponding DB in your TIA Portal application is named **HeraldVariables**, then the **S7 DB Name** should be populated with the exact same value, that is, **HeraldVariables**. Also, if the data-point within that Data-block is named, let's say **v3**, the **S7 Variable Name** field should be populated with that exact same value, that is, **v3**.

Figure 2.48 shows the overview section with one created Variable.



NAME	DESCRIPTION	ORGANIZATION	INTEGRATION	POINTER	CREATED ON	EDIT	DELETE
S7-1200 Variable	User manual PLC Variable	Devoted	Devoted PLC	"HeraldVariables".v3	11.10.2019 18:14		

Figure 2.48 – Overview section with one created Variable





### 2.12.1.3 Add new API Integration Variable

By clicking the Add button in the Add/Refresh section, the Add new Variable dialog appears. If the Integration dropdown is set to a previously defined API Integration, the UI will offer API specific setup as shown in Figure 2.49.

NAME	DESCRIPTION	ORGANIZATION	INTEGRA
S7-1200 Variable	User manual PLC Variable	Devoted	Devote

Figure 2.49 - Add new Variable dialog

The Save icon is disabled as long as values are not entered into mandatory fields. In this case, Variable name is the only mandatory field. Even though one can save a Variable with values entered only for mandatory fields, we strongly suggest that all values are entered since they are valuable for further setup and support. Once mandatory fields have values, the Save button becomes enabled. By clicking on the Save button, the user saves the Variable.

Figure 2.50 shows the overview section with one Direct Integration and one API Integration Variable

NAME	DESCRIPTION	ORGANIZATION	INTEGRATION	POINTER	CREATED ON	EDIT	DELETE
API Variable	User manual API Variable	Devoted	Devoted API	N/A	11.10.2019 18:18		
S7-1200 Variable	User manual PLC Variable	Devoted	Devoted PLC	"HeraldVariables".v3	11.10.2019 18:14		

Figure 2.50 – Overview section with one created Variable

## 2.12.2 Edit Variable

By clicking on the button in the Edit column, the corresponding Variable will be loaded into the Edit dialog for editing. If the user has insufficient permissions to edit a Variable, the edit button for the corresponding Variable will be disabled. Only administrator users can edit Variables.

Figure 2.51 shows the edit Variable dialog for the example Variable “API Variable” for the “Devoted API” Integration. Mandatory fields cannot be edited to be empty. The Save button becomes enabled once a change to any of the fields’ values is detected. At any time, the editing can be canceled by clicking on the Cancel button. The Variable’s assignment to an Integration cannot be changed (to do this, delete the first Variable, and add another one to the other Integration).

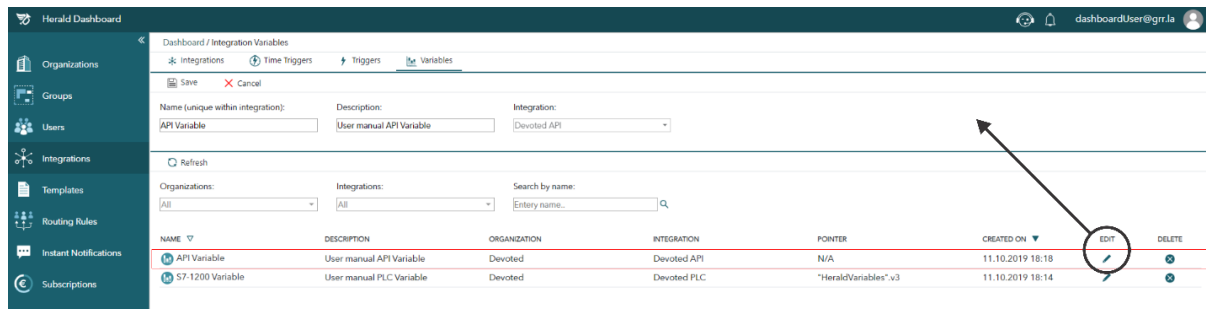


Figure 2.51 - Edit Variable dialog

## 2.12.3 Delete Variable

By clicking on the button in the Delete column, Herald services will initiate the deletion of the corresponding Variable. If the user has insufficient permissions to delete a Variable, the delete button for the corresponding Variable will be disabled. Only administrator users can delete Variables.

**The delete action is irreversible!** For that reason the user is required confirm the request by clicking on the 'Delete' button. By clicking on the delete button, the user initiates the deletion process. Since Variables can have dependencies such as assignment to Templates, the system will check what dependencies exist for the Variable the user is trying to delete. If no dependencies are found, the Variable will be deleted right away. If any dependencies are found, the user will be asked to confirm the delete action for the Variable and all of its dependencies by clicking on the 'Confirm delete' button. This will delete the chosen Variable and unassign all listed dependencies. The delete action can be aborted by clicking on the 'Cancel' button.

## 2.13 Templates

### 2.13.1 About

In the Herald Dashboard environment, **Templates** are used to define the content of notifications which are to be sent to users when defined Triggers are triggered. Both static and dynamic notification content is supported. Static content does not change depending on the moment of the notification's creation (it is just plain text). Dynamic content will be evaluated into concrete values at the moment of the notification's creation (user and/or time dependent data such as names, measurements, variable values etc.).

### 2.13.2 Templates overview

All pages accessed from the left navigation pane (read 2.4, numbered item 3) except the Subscriptions page are organized in three sections:

1. Add / Refresh section,
2. Filter section,
3. Overview section.

Figure 2.52 shows the Templates overview page with the previously described sections.

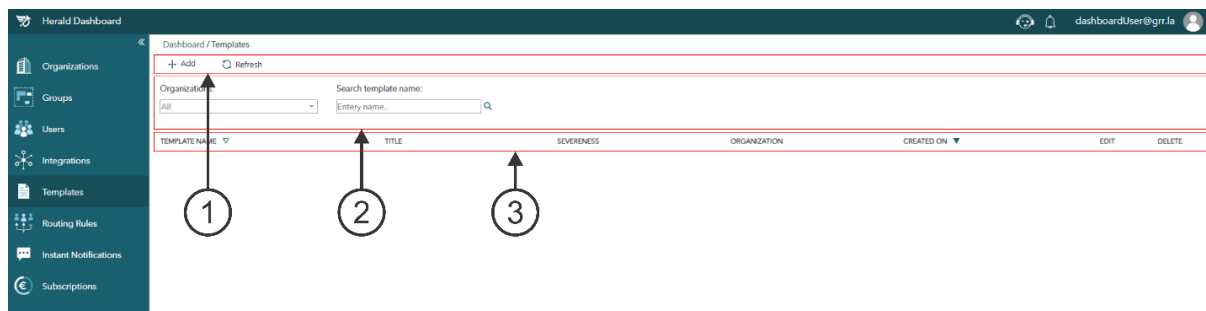


Figure 2.52 - Templates Overview

By default, all currently defined Templates will appear in the Overview section. Templates can be filtered by editing the Filter section. By clicking the Refresh button in the Add/Refresh section, the Overview section is reloaded with the currently set filter values. Templates can be filtered by their assignment to Organizations. Filtered Templates can be searched by their name. These options can be found in the Filter section.

In the overview section, Templates are displayed in a table where each Template is represented by one table row. Most of the table columns are dedicated for presenting the Template specific data (entered by the Template creator / editor). The last two columns on the right side of the table are so called action columns. By clicking on the icon in the column, the user initiates an action for the Template in the corresponding table row. This way, Templates can be Edited or Deleted.



## 2.13.3 Add new Template

By clicking the Add button in the Add/Refresh section, the Add new Template dialog appears (Figure 2.53).

NAME	VALUE	INTEGRATION	ADD
User first name	[User First Name]	Global	+
User last name	[User Last Name]	Global	+
User full name	[User Full Name]	Global	+
Organization name	[Customer Name]	Global	+

Figure 2.53 - Add new Template dialog

The Save icon is disabled as long as values are not entered into mandatory fields. In this case, Template name, Title and Content are mandatory fields. The Organization selection dropdown is preset to one of the previously added Organizations or, as in this case, to the pinned Organization. As previously explained, if an Organization has been pinned, all Organization selection dropdown throughout the entire Dashboard will be preset to this Organization and disabled for editing. The Template will be assigned to the Organization set in the Organization selection dropdown.

Using the Severeness dropdown the user can define the severeness of the future message. Three severeness options are available:

- Info (notifications will be colored in turquoise color in the Herald mobile applications),
- Warning (notifications will be colored in orange color in the Herald mobile applications),
- Error (notifications will be colored in red color in the Herald mobile applications).

As mentioned in the About section, Templates can feature both static and dynamic content. For dynamic content (Variables, User and Organization specific data), so called **Placeholders** are available and listed in the placeholder table (Figure 2.53, on the right side). Placeholders can be added to the Template's Title and Content. Placeholders can be introduced by drag & dropping them into the desired location or simply by clicking on the corresponding 'plus' symbol in the 'ADD' column of the placeholder table. When a placeholder is added it will be represented with the placeholder name enclosed in square brackets. When Triggers get triggered and as a consequence, notifications are generated, placeholders will be replaced with the appropriate content and as such sent to users. For example, if the [User First Name] placeholder has been used, it will be replaced with the corresponding user's real first name.

Now, [User First Name], [User Last Name], [User Full Name] and [Customer Name] are system placeholders which are available for all Templates. Integration specific placeholders like Variables will only become available if the user selects the corresponding Integration group within the Integration group dropdown selector. Once an Integration group is selected, Variables from that integration group will become available in the placeholder table.

Only system placeholders and placeholders from a single integration group can be used in one Template!

If a user changes the Integration group while creating the Template, used placeholders from the previously selected integration group will not be resolved in future notifications.



Even though one can save a Template with values entered only for mandatory fields, we strongly suggest that all values are entered since they are valuable for further setup and support. Once mandatory fields have values, the Save button becomes enabled. By clicking on the Save button, the user saves the Template.

Figure 2.48 shows the overview section with one created Template.

TEMPLATE NAME	TITLE	SEVEREITY	ORGANIZATION	CREATED ON	EDIT	DELETE
Example template	Template title	Warning	Devoted	14.10.2019 10:46		

Figure 2.54 – Overview section with one created Template

### 2.13.4 Edit Template

By clicking on the button in the Edit column, the corresponding Template will be loaded into the Edit dialog for editing. If the user has insufficient permissions to edit a Template, the edit button for the corresponding Template will be disabled. Operator and administrator users can edit Templates.

Figure 2.55 shows the edit Template dialog for the example Template “Example template”. Mandatory fields cannot be edited to be empty. The Save button becomes enabled once a change to any of the fields’ values is detected. At any time, the editing can be canceled by clicking on the Cancel button. The Template’s assignment to an Organization cannot be changed (to do this, delete the first Template, and add another one to the other Organization).

NAME	VALUE	INTEGRATION	ADD
S7-1200 Variable	[S7-1200 Variable]	Customer	+
User first name	[User First Name]	Global	+
User last name	[User Last Name]	Global	+
User full name	[User Full Name]	Global	+
Organization name	[Customer Name]	Global	+

TEMPLATE NAME	TITLE	SEVEREITY	ORGANIZATION	CREATED ON	EDIT	DELETE
Example template	Template title	Warning	Devoted	14.10.2019 10:46		

Figure 2.55 - Edit Template dialog



## 2.13.5 Delete Template

By clicking on the button in the Delete column, Herald services will initiate the deletion of the corresponding Template. If the user has insufficient permissions to delete a Template, the delete button for the corresponding Template will be disabled. Only administrator users can delete a Template.

**The delete action is irreversible!** For that reason the user is required to enter the exact name of the Template he/she would like to delete. Figure 2.56 shows that first step of the Template deletion process.

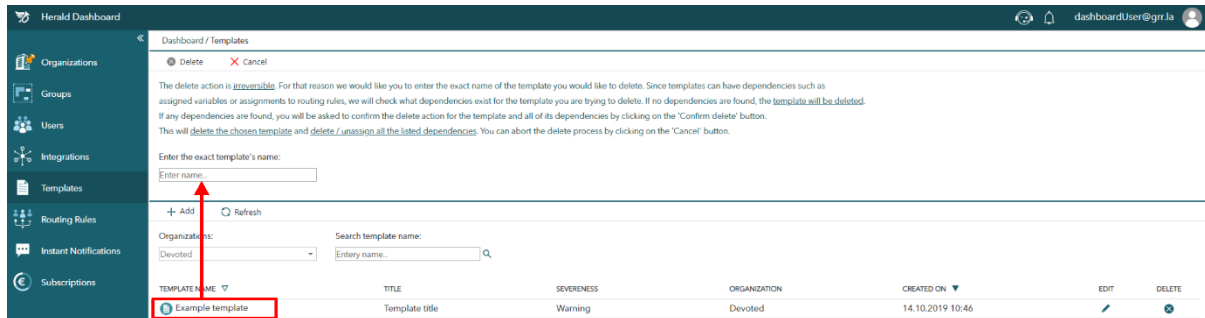





Figure 2.56 - Delete Template dialog

Once the exact name of the Template is entered, the Delete button becomes available. By clicking on the delete button, the user initiates the deletion process. Since Templates can have dependencies such as assigned Variables or assignments to Routing Rules the system will check what dependencies exist for the Template the user is trying to delete. If no dependencies are found, the Template will be deleted right away. If any dependencies are found, the user will be asked to confirm the delete action for the Template and all of its dependencies by clicking on the 'Confirm delete' button. This will delete the chosen Template and delete / unassign all the listed dependencies. The delete action can be aborted by clicking on the 'Cancel' button.

## 2.14 Routing Rules

### 2.14.1 About

In the Herald Dashboard environment, **Routing Rules** are used to define which users will receive which notifications when a Trigger gets triggered. Also, Routing Rules define notification mediums. Three notification mediums are supported:

-  App - users get alerted via the android mobile application,
-  Email - users get alerted via email on their defined email addresses and
-  SMS - users get alerted via SMS on their defined phone numbers.

In terms of user selection, Routing rules can be configured to include:

- all users of an Organization,
- specific users within an Organization and / or
- users filtered by their Roles and their assignments to Groups within the Organization.

### 2.14.2 Routing Rules overview

All pages accessed from the left navigation pane (read 2.4, numbered item 3) except the Subscriptions page are organized in three sections:

1. Add / Refresh section,
2. Filter section,
3. Overview section.

Figure 2.57 shows the Routing Rules overview page with the previously described sections.

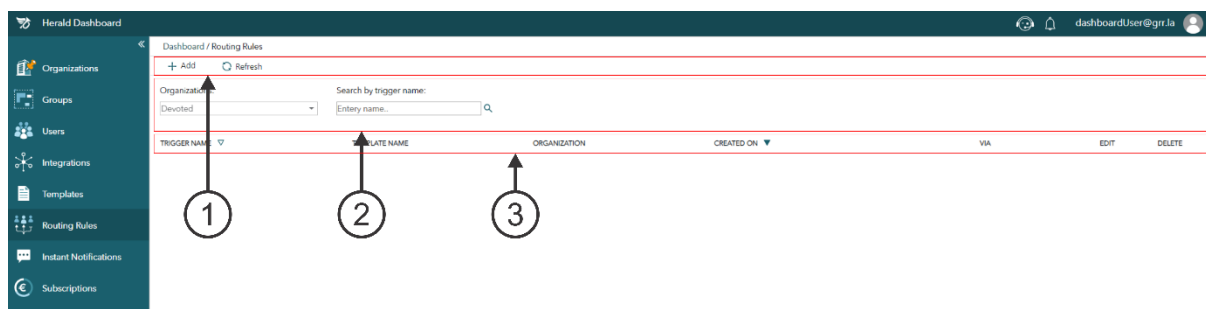


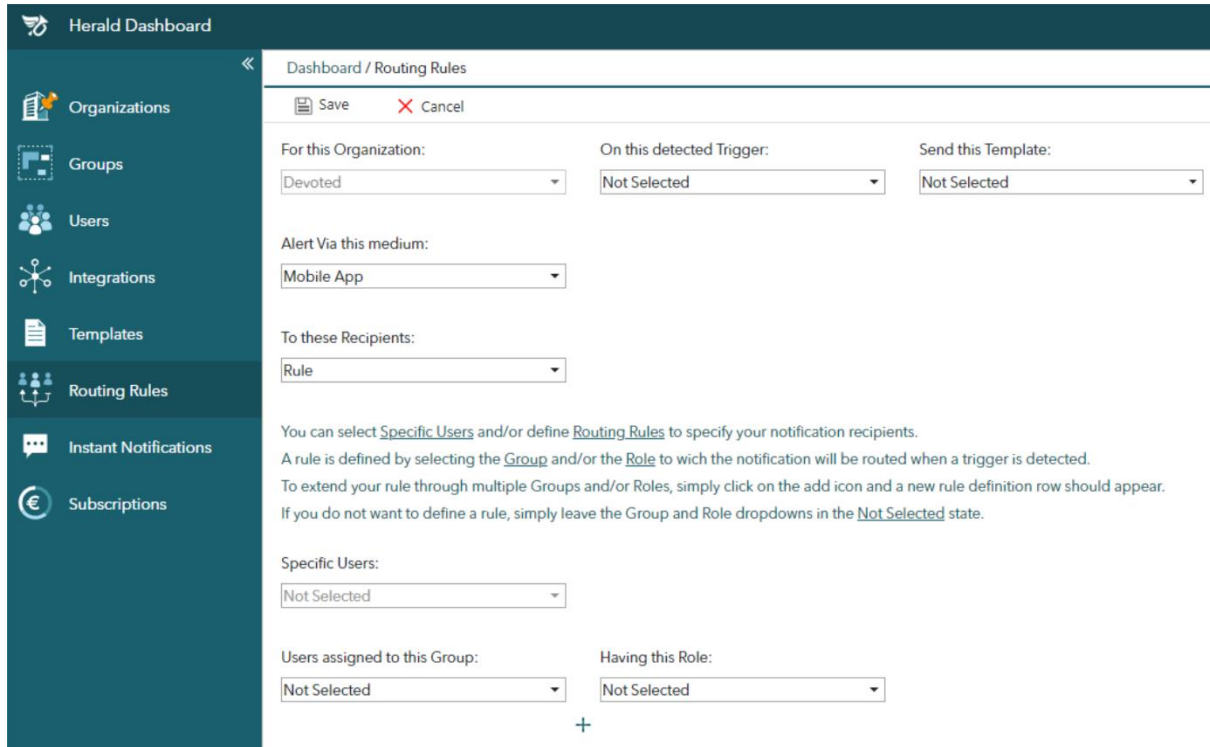
Figure 2.57 - Routing Rules Overview

By default, all currently defined Routing Rules will appear in the Overview section. Routing Rules can be filtered by editing the Filter section. By clicking the Refresh button in the Add/Refresh section, the Overview section is reloaded with the currently set filter values. Routing Rules can be filtered by their assignment to Organizations. Filtered Routing Rules can be searched by their name. These options can be found in the Filter section.

In the overview section, Routing Rules are displayed in a table where each Routing Rule is represented by one table row. Most of the table columns are dedicated for presenting the Routing Rule specific data (entered by the Routing Rule creator / editor). The last two columns on the right side of the table are so called action columns. By clicking on the icon in the column, the user initiates an action for the Routing Rule in the corresponding table row. This way, Routing Rules can be Edited or Deleted.

## 2.14.3 Add new Routing Rule

By clicking the Add button in the Add/Refresh section, the Add new Routing Rule dialog appears (Figure 2.53).



Dashboard / Routing Rules

Save Cancel

For this Organization: Devoted On this detected Trigger: Not Selected Send this Template: Not Selected

Alert Via this medium: Mobile App

To these Recipients: Rule

You can select [Specific Users](#) and/or define [Routing Rules](#) to specify your notification recipients.  
A rule is defined by selecting the [Group](#) and/or the [Role](#) to which the notification will be routed when a trigger is detected.  
To extend your rule through multiple Groups and/or Roles, simply click on the add icon and a new rule definition row should appear.  
If you do not want to define a rule, simply leave the Group and Role dropdowns in the [Not Selected](#) state.

Specific Users: Not Selected

Users assigned to this Group: Not Selected Having this Role: Not Selected

+

Figure 2.58 - Add new Routing Rule dialog

The Save icon is disabled as long as values are not entered into mandatory fields. In this case, the **Trigger**, **Template** and **Recipients dropdowns** are mandatory fields. The Organization selection dropdown is preset to one of the previously added Organizations or, as in this case, to the pinned Organization. As previously explained, if an Organization has been pinned, all Organization selection dropdown throughout the entire Dashboard will be preset to this Organization and disabled for editing. The Routing Rule will be assigned to the Organization set in the Organization selection dropdown.

Using the Trigger selection dropdown the user can define which Trigger needs to be triggered to activate this Routing Rule.

The Template selection dropdown is used to define the content of the notifications which are to be routed to users once the chosen Trigger gets triggered.

The medium selection dropdown specifies the mediums (app, email and/or SMS) through which users are going to be alerted when the specified Trigger gets triggered.

The recipients selection dropdown is used to define which users are going to get alerted. The user can choose to include all users or to define a user selection rule. If the rule option is chosen (Figure 2.53), additional options appear. The rule creator can select specific users. Additionally, the user can define user selection rules which will filter users according to their Roles and assignments to Groups.

To extend user selection rules through multiple Groups and/or Roles, simply click on the add icon and a new rule definition row should appear. If the Group and Role selection dropdowns are both set to “Not Selected” within a single user selection rule row, this row will be ignored. The logical operator between multiple user selection rule rows is “OR”. This means that multiple rows will, generally speaking, broaden the selection, not narrow it down.





Figure 2.59 shows the overview section with one created Routing Rule.

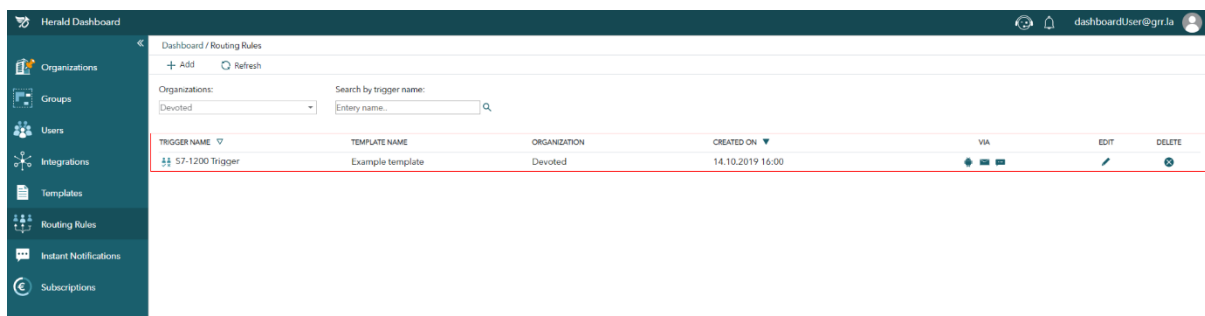


Figure 2.59 – Overview section with one created Routing Rule

## 2.14.4 Edit Routing Rule

By clicking on the button in the Edit column, the corresponding Routing Rule will be loaded into the Edit dialog for editing. If the user has insufficient permissions to edit a Routing Rule, the edit button for the corresponding Routing Rule will be disabled. Operator and administrator users can edit Routing Rules.

Figure 2.60 shows the edit Routing Rule dialog for the previously created example Routing Rule. At any time, the editing can be canceled by clicking on the Cancel button. The Routing Rule's assignment to an Organization cannot be changed (to do this, delete the first Routing Rule, and add another one to the other Organization).

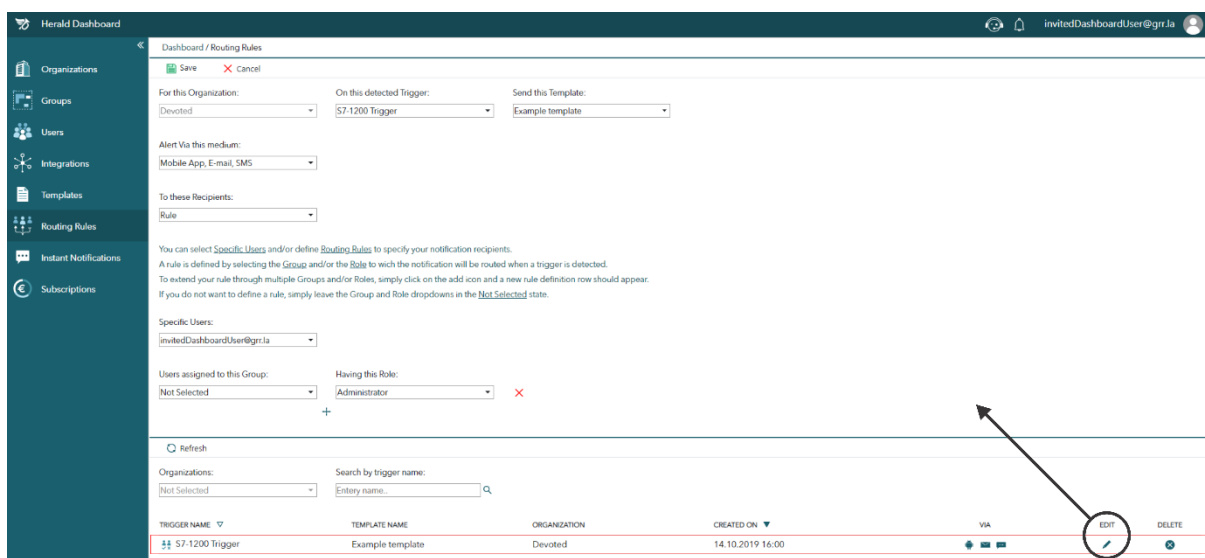


Figure 2.60 - Edit Routing Rule dialog

## 2.14.5 Delete Routing Rule

By clicking on the button in the Delete column, Herald services will initiate the deletion of the corresponding Routing Rule. If the user has insufficient permissions to delete a Routing Rule, the delete button for the corresponding Routing Rule will be disabled. Only administrator users can delete Routing Rules.

**The delete action is irreversible!** For that reason the user is required confirm the request by clicking on the 'Delete' button. By clicking on the delete button, the chosen Routing Rule will be deleted. The delete action can be aborted by clicking on the 'Cancel' button.



## 2.15 Instant Notifications

### 2.15.1 About

Sometimes it is useful to send one-time notifications or notifications which contain information that might not be valid or important at some other point in time. To address these situations, Herald Dashboard supports sending of so called Instant Notifications where the user can select the Organization, Organization Users as recipients, define the Notification Severeness and delivery medium, along with the notification Title and Content.

### 2.15.2 Send new Instant Notification

Figure 2.61 shows the 'send new Instant Notification' dialog.

NAME	VALUE	INTEGRATION	ADD
User first name	[User First Name]	Global	+
User last name	[User Last Name]	Global	+
User full name	[User Full Name]	Global	+
Organization name	[Customer Name]	Global	+

Figure 2.61 – Send new Instant Notification

As previously explained, the user can select an Organization, choose notification recipients from the available Organization Users, select the notification's Severeness level and define delivery mediums (app, email and/or SMS).

After editing mentioned fields and adding the notification Title and Content, the Send button becomes available. Adding placeholders to notification Title and Content is supported. Since no Integration specific data is available, only system placeholders are available (read 2.13.3).

By clicking on the 'Send' button, the user initiates the sending of instant notification(s).

#### 2.15.2.1 Subscription handling

Billing for sent Instant Notifications is handled through the Instant Notification Subscription. By default, every new Organization gets a **Demo Instant Notification Subscription** which can be used for testing purposes since it allows sending of up to 10 instant notifications for free. After this, the subscription will become usage deactivated.

The Demo Instant Notification Subscription can be upgraded to the **Consumption Instant Notification Subscription**. Under this subscription, an unlimited amount of instant notifications can be sent provided that there are enough resources on the Organization's account. The pricing for the Consumption Instant Notification subscription can be viewed in the [Pricing Policy Documentation](#).

## 3 SUBSCRIPTIONS

### 3.1 About

All billing and payments for services provided by the platform are handled through Subscriptions. The main Herald Dashboard entity to which all Subscriptions are coupled is the Herald Organization.

The platform supports a prepaid billing model only. This means that the Organization's administrator users must add resources to their Organization's Herald account before the Organization can start using any platform service under a paid (non-Demo) subscription. For now, two payment options are supported:

- Online payment (via [Raiffeisen Bank](#)'s official electronic payment integration [e-pay](#)),
- Payment via Bank (all necessary data displayed in the corresponding section).

When a payment is successfully processed, the Organization's Herald account balance will be increased with the paid amount. The Organization's account balance will simply be referred to as the account balance.

Depending on the configured Subscriptions and usage volume, the account balance will decrease over time. Balance subtractions will be made based on the Organization's usage of the services provided by the platform. All logged billable events/services provided from the platform to the Organization will automatically subtract the corresponding amount from the account balance. Details on the billing model and Subscription prices can be viewed in the [Pricing Policy Documentation](#).

The Subscriptions section shows all the available Subscriptions for all of the Organizations the logged in user administers. Only administrator users can visit this section. In case, the user administers multiple Organizations, Organization tiles will be displayed to choose for which Organization the user wants to view the available Subscriptions. If the user administers only one Organization, a click on left navigation pane's (read 2.4, numbered item 3) 'Subscriptions' link will lead the user directly to the corresponding Organization's subscriptions overview.

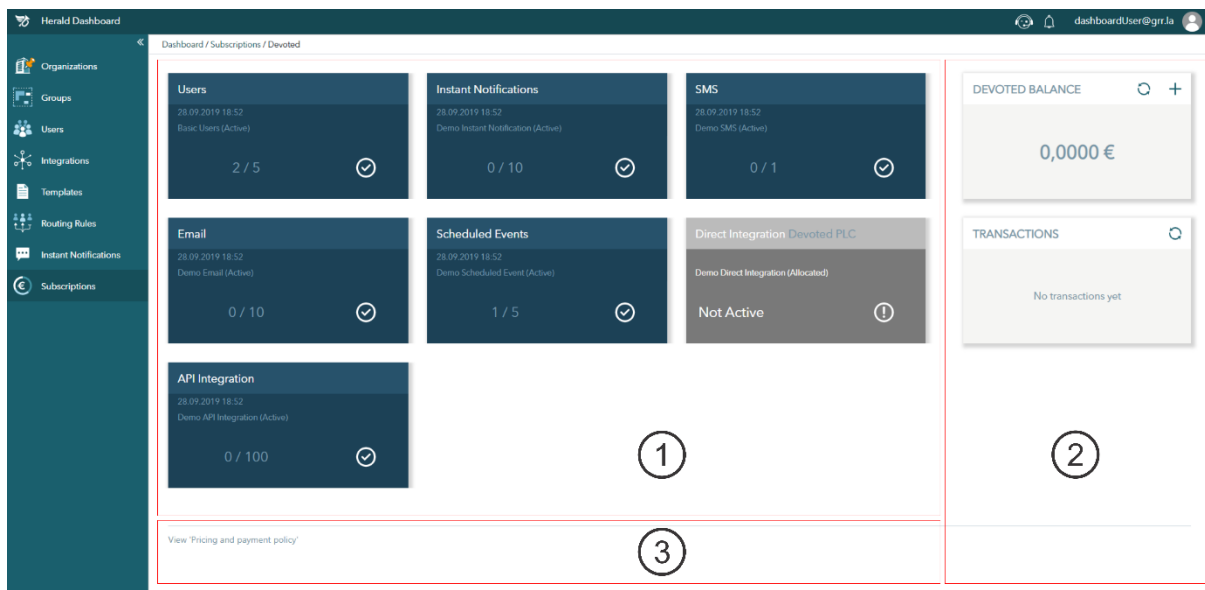


Figure 3.1 - Subscriptions overview for example Organization

Figure 3.1 shows the subscriptions overview for the example Organization 'Devoted'. The subscriptions overview page has three main sections:

1. Subscriptions overview,
2. Balance and transactions overview,
3. Pricing and payment policy link (opens the [Pricing Policy Documentation](#) in a new tab).

## 3.2 Subscriptions overview

In the Subscriptions overview section, all currently available subscriptions with their statuses are displayed in form of tiles. The tile header holds the subscription name. Right beneath, the subscription status change date-time is displayed. Below that, the current subscription option and its status in curly brackets are shown.

The lower right section of a tile features a status related icon. If the current subscription option is active, the icon will be a check symbol. Otherwise, the icon will be an exclamation mark, drawing the user's attention to check whether everything is ok with the corresponding subscription.

Tile colors help to identify the status of a subscription more easily. Active subscriptions are displayed in dark petroleum color while deactivated or generally speaking, not active subscriptions are shown in a gray color. In this case, the Direct Integration subscription is not active and thus displayed in gray.

The lower left section of a tile shows the current consumption status for the corresponding subscription. For example, the figure below shows that currently 2 of 5 available users have been assigned to the Organization.

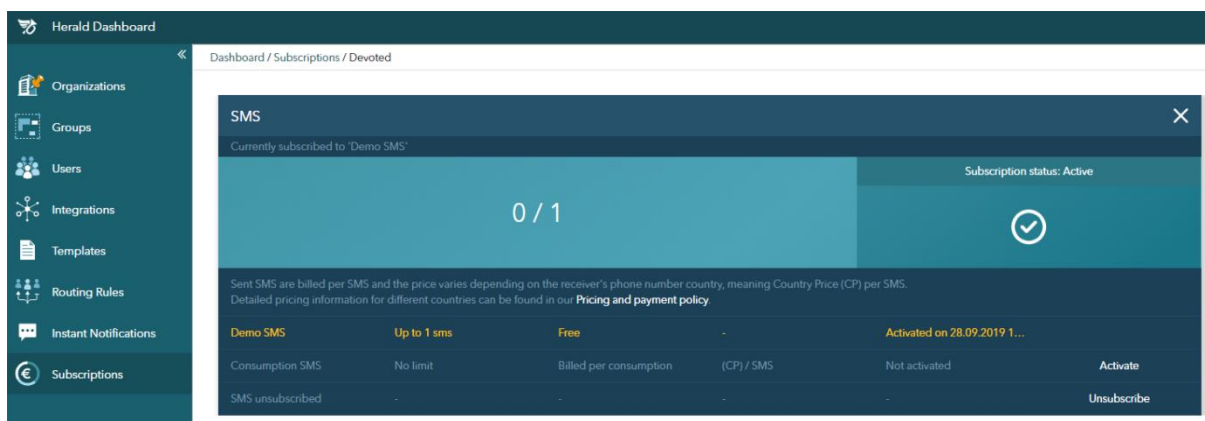


Figure 3.2 – Expanded SMS subscription tile

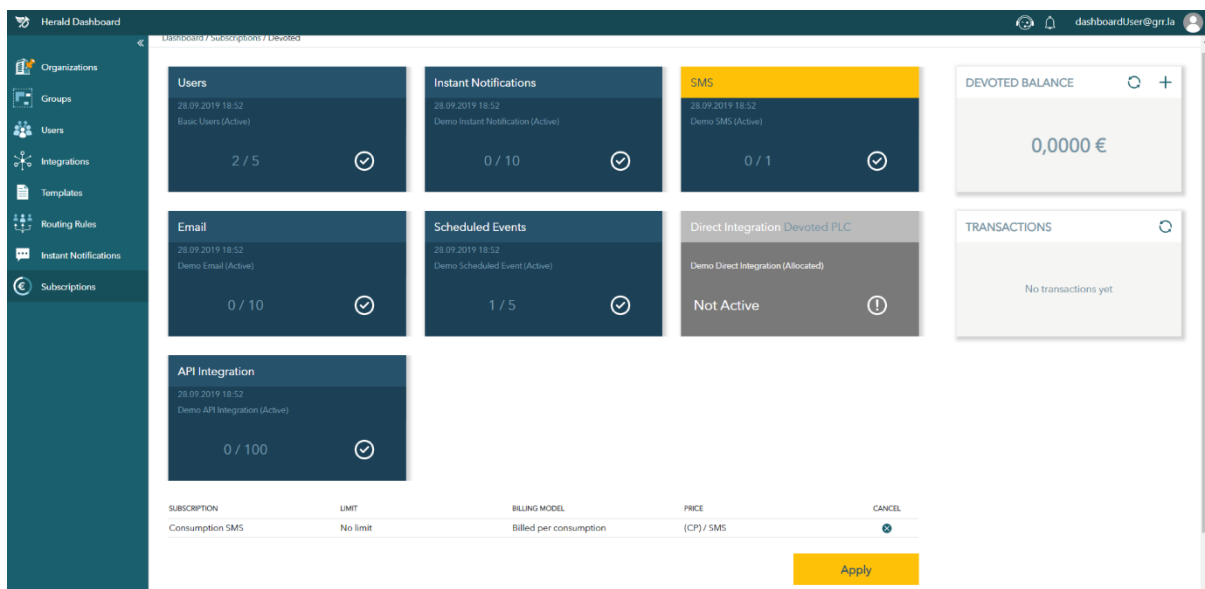


Figure 3.3 – Apply subscription changes table

By clicking on a tile, the corresponding subscription's tile expands. The expanded tile shows all the relevant data for the corresponding subscription. Figure 3.2 shows the expanded SMS subscription tile. The expanded tile shows the currently active subscription option and gives details on pricing regarding the subscription in general. All data which was shown on the collapsed tile is now also displayed, just in more detail.



The expanded tile shows all the available subscription options for the corresponding subscription. The current subscription option is highlighted in orange color. In this case, the Devoted Organization is currently subscribed to the 'Demo SMS' SMS subscription option which allows one SMS to be sent free of charge.

Current subscription options can be changed by clicking on the Activate button next to available alternative subscription options. In this case, the 'Consumption SMS' SMS subscription option is available. By clicking on the 'Activate' button, the requested change is loaded into an apply table. Figure 3.3 shows the apply-table with the requested subscription option change listed as one table row. As you can see, the corresponding tile's header is highlighted in orange to show that a change has been requested for that subscription. Changes are applied by clicking on the apply button below the apply-table. To make subscription upgrades (go from Demo subscription option to a Consumption subscription option), a minimum Organization balance of 10 € is necessary. Changes can be canceled both from the expanded tile and from the apply-table. Some subscriptions can be totally unsubscribed. If this option is available, it will be listed as one of the subscription options in the expanded subscription tile (Figure 3.2, lowest displayed option).

### 3.3 Balance overview

As previously explained, the subscriptions overview is consists of three section, one being the balance overview section (Figure 3.1, number 2). Figure 3.4 shows the Balance overview section for an example Organization called 'Arctic Cave'.

The upper (balance) tile shows the Organization's name and the Organization's current Balance. Four decimal points are displayed intentionally to allow the user to follow the withdrawal of resources even for the smallest billed amounts (single email for example). Roughly said, this Organization's current balance is at 67.2 €. In the upper right section of the balance tile, two buttons are available. The refresh button will reload and refresh the balance tile. This might be necessary if the user wants to check for changes to the Organization's balance while on the subscriptions overview section. A click on the 'plus' button will open a new tab for adding resources to the Organization's balance. This is explained in 3.3.1.

The lower (transactions) tile shows the Organization's transaction history grouped by dates. This means that all transaction changes (withdrawals and additions) are grouped on a daily basis and displayed in this section.

Individual transactions within a day can be viewed by clicking on the corresponding table row. This will open an overview of all transactions for the corresponding day while hiding grouped daily transactions. In that overview, details are shown such as transaction date-time, transaction direction, transaction amount as well as subscription (product) association based on which withdrawals have been made. A click on the refresh button will reload and refresh the transactions tile.

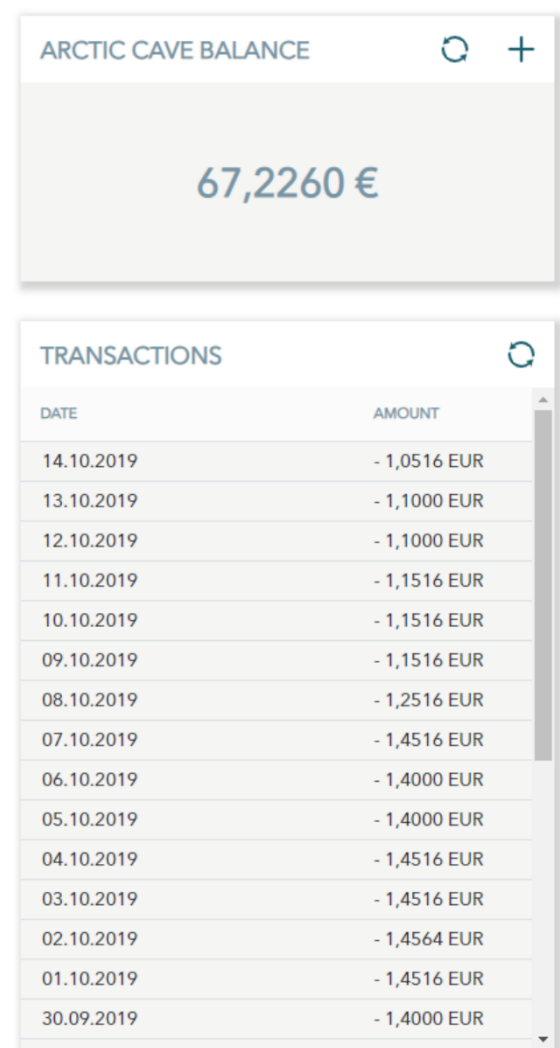


Figure 3.4 - Balance overview section

## 3.3.1 Add to Organization's balance

A click on the 'plus' button in the balance tile will open a new tab for adding resources to the Organization's balance. For now, two payment options are supported:

- Online payment (via [Raiffeisen Bank](#)'s official electronic payment integration [e-pay](#)),
- Payment via Bank (all necessary data displayed in the corresponding section).

For payments via Bank, all data necessary is given in the 'ADD TO BALANCE via Bank' section. Here, the beneficiary bank, SWIFT code, bank address, IBAN code etc. can be found. If you should need any additional data, please contact our [support](#). Also, to assure a quick update of your organization's Herald balance, please write to our [support](#) that you have initiated a payment so that we can check for its arrival and update your balance as soon as possible. If you choose this method of payment, please note that transfer costs will most probably be applied if you are initiating the payment form outside of Bosnia and Herzegovina. In this case, the suggested payment option would be the online payment option.

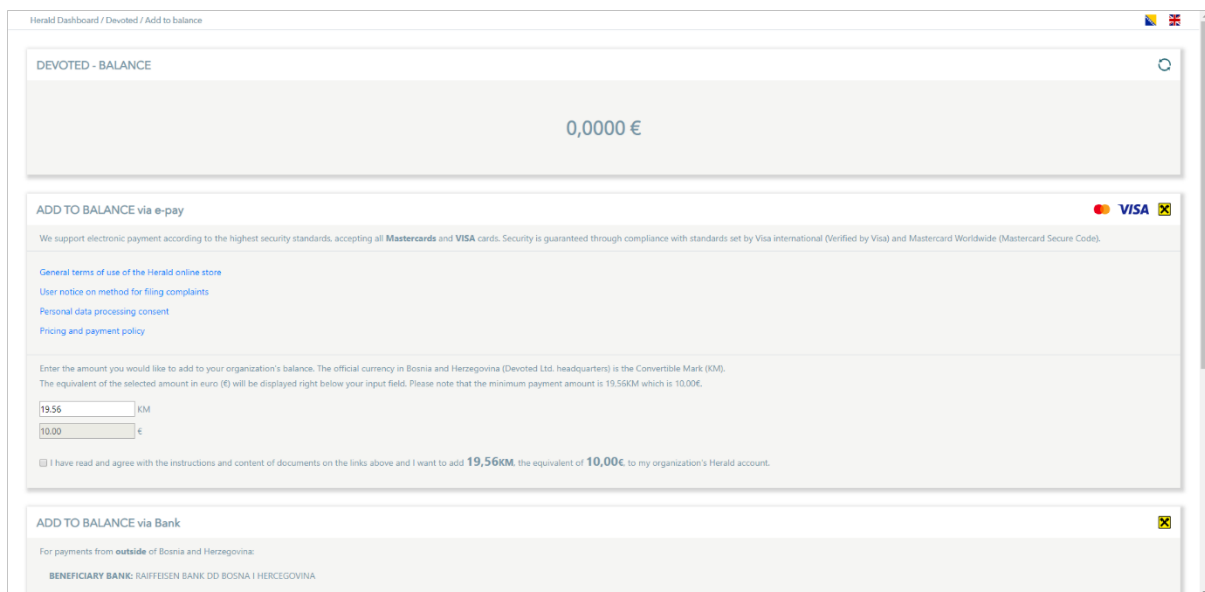


Figure 3.5 - Add to balance tab

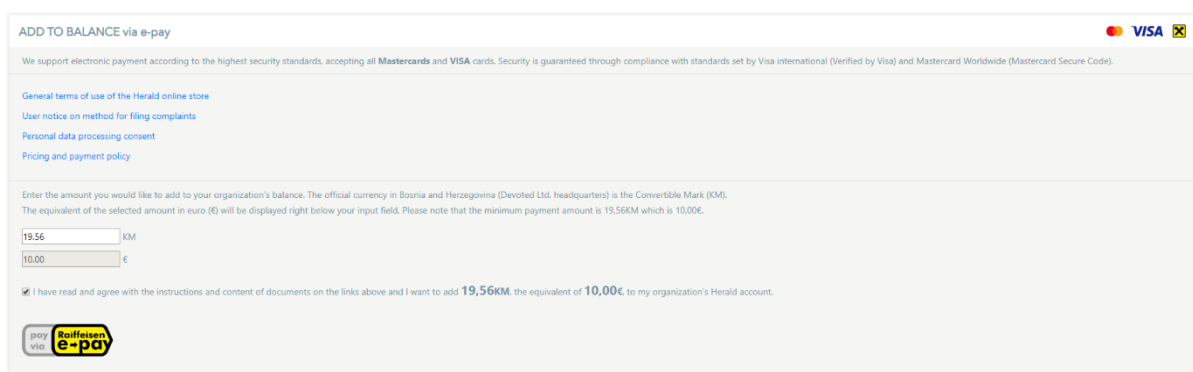


Figure 3.6 – Initiate Pay via e-pay transaction



Figure 3.7 – Finish Pay via e-pay transaction

Figure 3.5 shows a screen shot of the opened new tab which opens when the User clicks on the ‘plus’ button in the balance tile. Within the new tab, three tiles are available. The upper tile shows the Organization’s current balance. At the bottom, the user can find all data necessary to make a payment via Bank.

The middle tile is used to add to the Organization’s balance via e-pay. The official currency in Bosnia and Herzegovina (Devoted Ltd. headquarters) is the Convertible Mark (KM). We are required by law to process payments in our domestic currency which is why the User must enter the desired amount in KM. The equivalent of the selected amount in euro (€) will be displayed right below the input field for KM.

Generally speaking, the payment procedure is fairly simple and includes just a few steps:

1. Read all documents provided via links (colored in blue in Figure 3.6),
2. Enter the amount to add to the Organization’s balance in Convertible Mark (KM). The equivalent in euro (€) will be shown right below the input field,
3. Click on the checkbox to give payment consent and confirm agreement with the content of the documents from point 1. This will initiate the transaction and shortly, the ‘pay via e-pay’ button should appear (Figure 3.6),
4. A click on the ‘pay via e-pay’ button opens the [Raiffeisen Bank](#)’s official electronic payment integration [e-pay](#) page (Figure 3.7) where the user can enter the card data and finish the transaction,
5. Enter card data and click on the ‘Pošalji/Send’ button to finish the transaction.

If the finished transaction is accepted for processing, a success message should be displayed indicating that the transaction will be processed soon. As soon as the transaction is processed, the paid amount should appear on the Organization’s balance.

Otherwise, a failure message will be displayed indicating that something went wrong. In this case, please check your card status and balance or contact our [support](#) to assist you.



## 4 CONCLUSION

This document gives the Herald Dashboard users explanations on the supported functionalities and instructions on how to utilize them to best serve the needs and requirements specific to their Organizations.

For further support, custom requests or suggestions on improving this document or the product in general, please contact our [support](#).

Thank you for choosing Herald

[www.devoted.ba](http://www.devoted.ba)

May Herald serve you well!