



zapy

Documentation

Version 1.0
September 2014



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

contents

1. what is zapy?
 - 1.1. illustration
 - 1.2. example of use
2. inputs
3. outputs
4. installation
5. control panel/settings
 - 5.1. control panel I/O
 - 5.2. control panel general settings
 - 5.3. control panel QR-Code/NFC
 - 5.4. control panel users
 - 5.5. control panel log
 - 5.6. control panel license
6. input actions in detail
7. output actions in detail
8. remote application (HTML/CSS/JS)
 - 8.1. remote application HTML file
 - 8.2. remote application CSS file
 - 8.3. remote application JS file
 - 8.4. remote application HTML exit file
9. license / costs



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

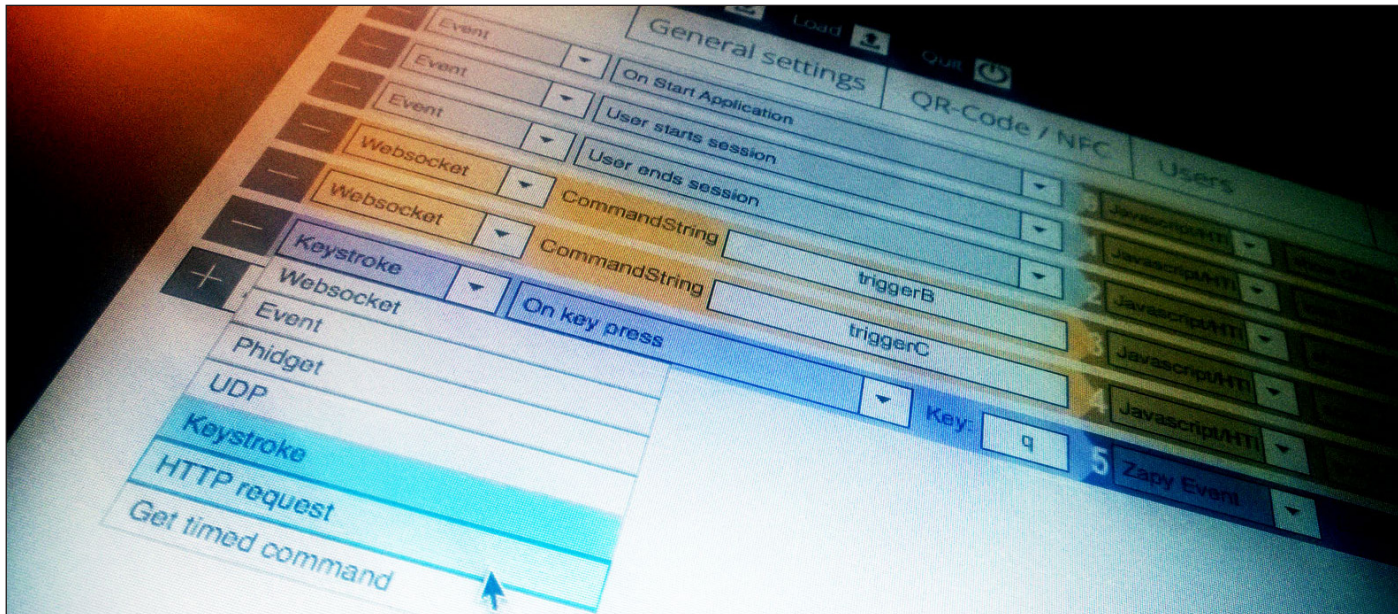
info@dissenter.ch
www.dissenter.ch

1. What is zapy

zapy is a standalone software which enables a connection of various devices and applications.

zapy uses different common technologies and protocols like websockets, UDP, http and more to communicate. For example it may be used to control an application via smartphone. Or it can be used to switch power sources by motion sensors etc.

You may specify various inputs by just selecting them in a dropdown list. Then you tell **zapy** what should happen next.



zapy runs on customer's smartphones without installation of any app.

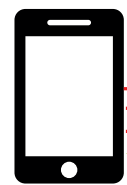


Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

1.1. Illustration of zapy

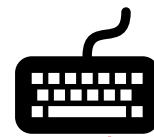
Smartphone



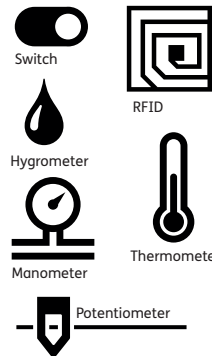
2. Connection via QR code reader or by entering an individual code or NFC

zapy communicates via websockets. No installation is required. The remote application is browser-based. Just an internet connection is required.

Keyboard

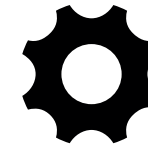


Sensors



... etc. ...
different analog and digital inputs via phidgets

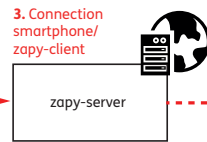
Application



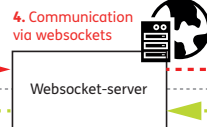
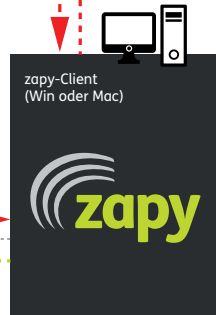
zapy may process inputs from an individual application



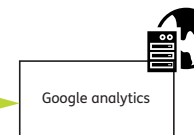
1. Generation of individual QR codes



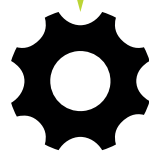
3. Connection smartphone/ zapy-client



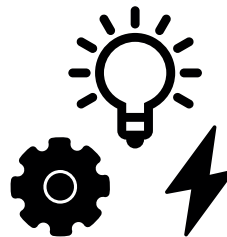
4. Communication via websockets



UDP-capable digital signage player (for example BrightSign)



control of an individual application



Using phidgets servo motors, stepper motors, relays and much more may be controlled



zapy may display web contents in fullscreen. They may be easily controlled by javascript.



zapy may control various standard devices via UDP or websockets.

Digital signage

Application

Motion

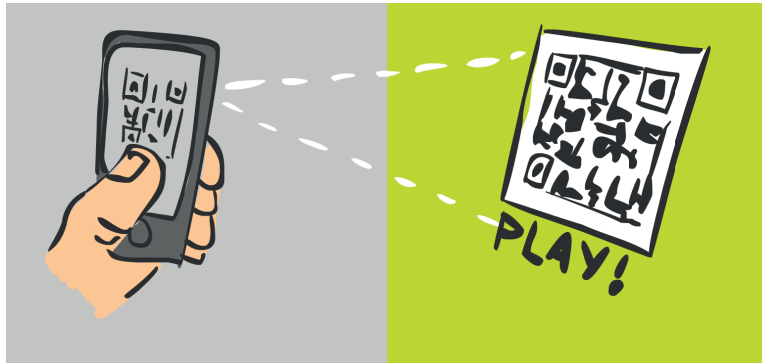
Webcontent

...etc...



1.2. Example of use

Scan the QR-code on the shopping window to use zapy on your smartphone.
Customer will be invited to visit the shop or the online-store of the advertiser.



Attract attention

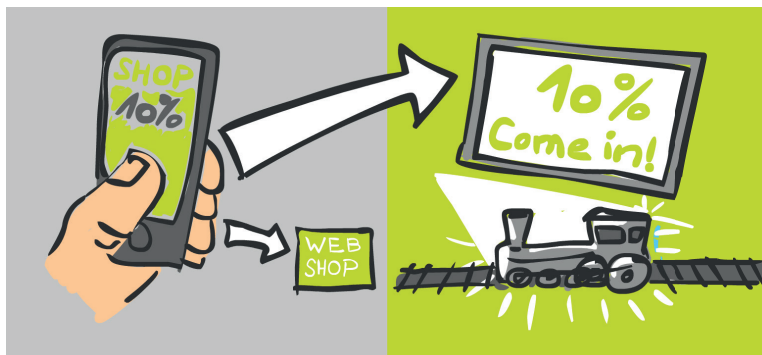
The shop window promises the customers added value by scanning the QR code. This can be a discount or an appeal to the play instinct.



Interaction

Customers may interact through the window with the product.

On the smartphone of the user a product-specific control appears.



Call to action

After a defined time or by reaching a goal (eg playful application) customers are called to action. For example by displaying a discount voucher on their smartphone to use in the shop.



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

No limits to your **ideas**.



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

2. Inputs

zapy can process different input sources:

- Websockets
- UDP
- HTTP-Requests
- Phidgets
- Keystrokes
- specific Events




3. Outputs

zapy can translate commands to various output types

- Websockets
- UDP
- HTTP-Requests
- Phidgets
- JS
- SWF-functions
- specific Actions





make your product
or service
interactive.



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

4. Installation

1. Visit <http://www.zapy.ch/downloads.html> and download **zapy**.
2. Install the application.
3. If you'd like to use phidgets visit http://www.phidgets.com/docs/Operating_System_Support and download the driver required.
- 3a. Open the Phidgets control panel and start the webservice. Set startup type to ,Automatic'.
4. Run zapy application as administrator (Windows) or just run it (Mac)

zapy offers the full scope of functions in the unlicensed version. The unlicensed version shuts down every 10 minutes and must be restarted manually. For a license open the control panel (see chapter 5) and select ,license'. More information about licencing zapy at chapter 6.



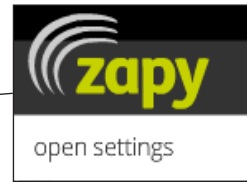
Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

5. zapy control panel

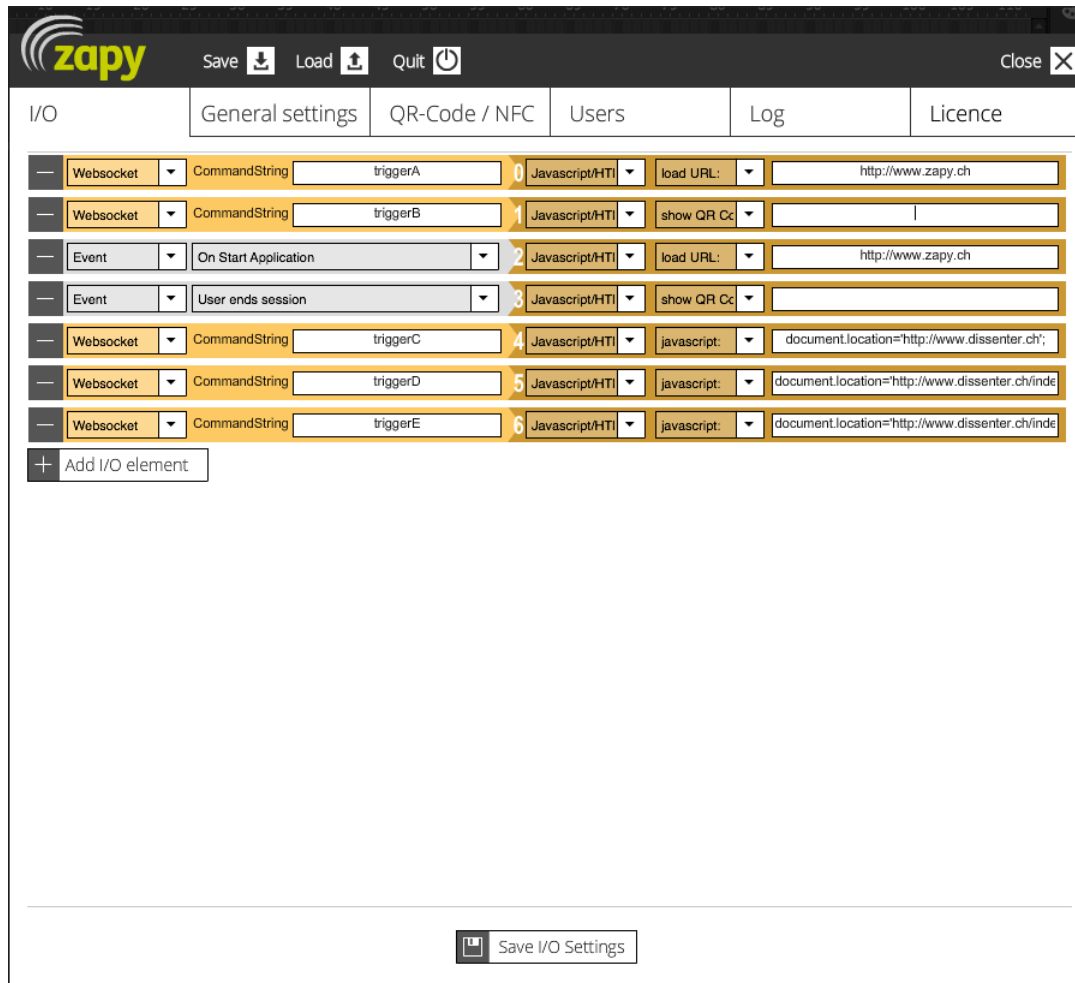
To open the zapy control panel

- press ,F1‘
- click on ,open settings‘ (only when swf-view and HTML-view disabled)
- click in the left lower corner of the screen (only if swf-view enabled)



5.1. Control panel I/O view

In the I/O view you define what happens at which input action. Click on ‚Add I/O element‘ to add a new action. Select on the left side the input and on the right side the output action. More information about inputs/outputs in chapter 7.



The screenshot shows the Zapy I/O control panel interface. At the top, there is a header bar with the Zapy logo and navigation buttons: Save, Load, Quit, and Close. Below the header, there are tabs for 'I/O', 'General settings', 'QR-Code / NFC', 'Users', 'Log', and 'Licence'. The main area contains a list of I/O elements, each with a minus sign on the left, a dropdown menu for the input type, a text field for the command string, a number in a circle, a dropdown menu for the output type, and a text field for the output action. The elements are:

- 0: Websocket, CommandString: triggerA, Javascript/HTI, load URL: http://www.zapy.ch
- 1: Websocket, CommandString: triggerB, Javascript/HTI, show QR Co
- 2: Event, On Start Application, Javascript/HTI, load URL: http://www.zapy.ch
- 3: Event, User ends session, Javascript/HTI, show QR Co
- 4: Websocket, CommandString: triggerC, Javascript/HTI, javascript: document.location='http://www.dissenter.ch';
- 5: Websocket, CommandString: triggerD, Javascript/HTI, javascript: document.location='http://www.dissenter.ch/inde
- 6: Websocket, CommandString: triggerE, Javascript/HTI, javascript: document.location='http://www.dissenter.ch/inde

At the bottom left, there is a button labeled '+ Add I/O element'. At the bottom center, there is a button labeled 'Save I/O Settings'.



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

5.2. Control panel General settings

In the general settings you specify the main settings for your application.

The screenshot shows the Zapy control panel with the following settings:

- I/O**: Save, Load, Quit, Close
- General settings**:
 - Display SWF-File: Select file SWF: **Example3.swf**
 - Display HTML-File: URL: **file:///Users/severin/Dropbox/Projekte/zapy/http_docs/test.htm** Save
 - Websockets: Latency: **165 ms**
 - Max. session time in seconds (Terminates a session automatically if no other event forced cancellation): **600** Save
 - URL GUI-HTML: **http://www.zapy.ch/remotes/zapydefault/ws.html** Save
 - URL Websocketsserver: **ws://achex.guzu.ca:4010** Save
 - Recieve UDP: IP: **192.168.0.21**
 - UDP-Port: **9003** Save
 - Act as Webserver: IP: **192.168.0.21** Address to display QR Code: **http://192.168.0.21:9004/qr/index.html**
 - Webserver-Port: **9004** Save
 - Analytics Category: **GATest** Save
 - Google Analytics ID: **UA-51794141-1** Save

activate ,Display SWF-File' to load and display an swf-application which may be controlled by zapy actions.

activate ,Display HTML-File' to load and display a HTML-file which may be controlled by Javascript. Enter the URL of start HTML-file. Include `` in the HTML-file which will be dynamically replaced by the QR code image.

activate none of them to run zapy in application mode.

activate to use websockets.

enter the session maximum time.

enter the URL to your GUI-HTML-file (remote application) which will be displayed on the smartphone when scanning the QR code.

This is the default websocket server address. Do not change if not really necessary

activate to enable zapy to receive UDP commands

enter the UDP port

activate to run zapy as webserver

enter the webserver port

Enter your google analytics category and code to

track events in google analytics



5.3. Control panel QR-Code/NFC

This window let you specify the behavior and the appearance of your QR Code.

The screenshot shows the Zapy control panel for QR-Code/NFC settings. The window title is 'Documentation_1_0.indd @ 76 %'. The menu bar includes 'Save', 'Load', 'Quit', and 'Close'. The main menu has tabs for 'I/O', 'General settings', 'QR-Code / NFC', 'Users', 'Log', and 'Licence'. The 'QR-Code / NFC' tab is active. The settings are as follows:

- Dynamic QR Code:** A radio button is selected, with a 'Save' button next to it.
- Save QR Code as image to disk:** A 'Save' button is present.
- Allow user queue:** A radio button is selected.
- Address for NFC Tag:** The value is 'n/a'.
- QR-Code X-Position:** Input field with '380' and a 'Save' button.
- QR-Code Y-Position:** Input field with '150' and a 'Save' button.
- QR-Code Size:** Input field with '900' and a 'Save' button.
- QR-Code Size:** Input field with the text 'Scannen Sie den QR-Code oder öffnen
Sie www.zapy.ch und geben Sie '@Cod' and a 'Save' button.
- QR Orientation:** A control with a square icon and an upward arrow.
- QR Transition in:** A dropdown menu with 'Cube' selected.
- QR Transition out:** A dropdown menu with 'Cube' selected.
- Show transition:** A 'Show transition' button.

Select if the QR Code should be dynamic (changes each successful connection) or static (for example to print out a connection code)

If QR Code is static you may save an image of it

Specify if there will be one user at once - or if users may connect and then displaying them a waiting message.

Choose static QR Code to get an address for NFC Tag.

Design options for QR Code. You may change size, position, orientation and transition. Enter the message which will be displayed in the QR Code. use '@Code' as a placeholder for the generated connection code. Transition only takes effect when application runs in swf-mode.

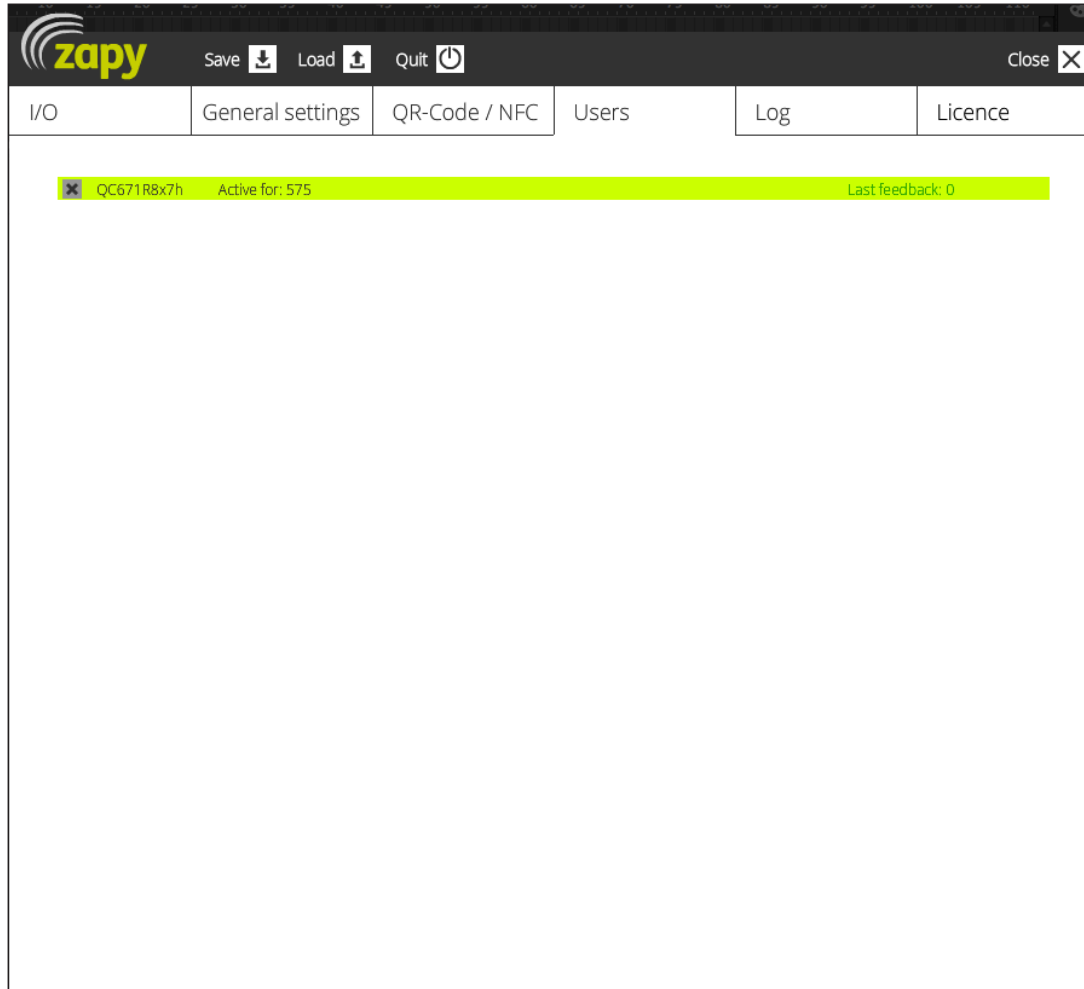


Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

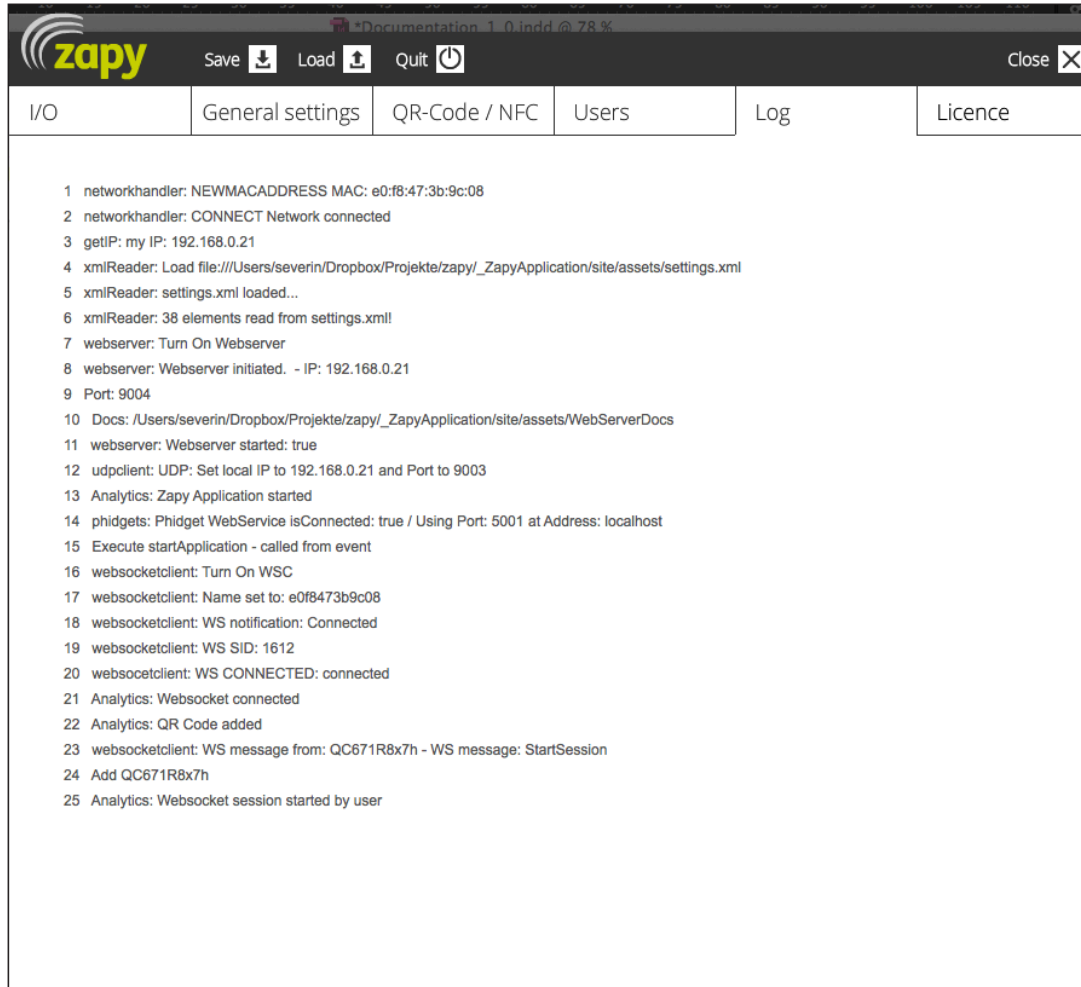
5.4. Control panel Users

Overview of active and waiting (if allow user queue is on) users.



5.5. Control panel Log

The log file of zapy application. If a problem occurs please have a look at the log first. Sometimes it may tell you what went wrong.



```
1 networkhandler: NEWMACADDRESS MAC: e0:f8:47:3b:9c:08
2 networkhandler: CONNECT Network connected
3 getIP: my IP: 192.168.0.21
4 xmlReader: Load file:///Users/severin/Dropbox/Projekte/zapy/_ZapyApplication/site/assets/settings.xml
5 xmlReader: settings.xml loaded...
6 xmlReader: 38 elements read from settings.xml!
7 webservice: Turn On Webservice
8 webservice: Webservice initiated. - IP: 192.168.0.21
9 Port: 9004
10 Docs: /Users/severin/Dropbox/Projekte/zapy/_ZapyApplication/site/assets/WebServerDocs
11 webservice: Webservice started: true
12 udpclient: UDP: Set local IP to 192.168.0.21 and Port to 9003
13 Analytics: Zapy Application started
14 phidgets: Phidget Webservice isConnected: true / Using Port: 5001 at Address: localhost
15 Execute startApplication - called from event
16 websocketclient: Turn On WSC
17 websocketclient: Name set to: e0f8473b9c08
18 websocketclient: WS notification: Connected
19 websocketclient: WS SID: 1612
20 websocketclient: WS CONNECTED: connected
21 Analytics: Websocket connected
22 Analytics: QR Code added
23 websocketclient: WS message from: QC671R8x7h - WS message: StartSession
24 Add QC671R8x7h
25 Analytics: Websocket session started by user
```



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur


info@dissenter.ch
www.dissenter.ch

5.6. Control panel License

If you like zapy and want to use it please fill out the license form within the application. The unlicensed version will shut down every couple of minutes. Please be aware that the license will only be valid on the computer you filled out the form. See details about license costs at chapter 6.

zapy application by: Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur
Switzerland

info@dissenter.ch
www.dissenter.ch
Tel. +41 (0)52 202 01 85
Fax. +41 (0)52 202 01 87

 DISSENTER

Connect, control, **interact.**



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

6. Input actions in detail

There are different input actions zapy may process. You may specify them in the I/O control panel.

• **Websocket**

Enter in the textfield 'CommandString' the specific string that you will send from your remote application.

• **Event**

Events are zapy application-events like 'start application' and so on. Choose an event to execute an output action at specific events of the zapy application.

The events:

- On Start Application: Triggers once on application starts
- User starts session: Triggers each time a user connects via QR Code / NFC
- User ends session: Triggers each time a user drops connection
- Ready for user to connect: Triggers once when the QR Code appears and application is ready
- No internet connection: Triggers on connection failure

• **Phidget**

Processes inputs from analog or digital phidget sensors

- Analog
 - > If value gets greater than: Triggers each time the value gets greater than the value specified
 - < If value gets smaller than: Triggers each time the value gets smaller than the value specified
 - - pass value to output (@val): Triggers each time the value changes and passes the new value as ',@val'-variable to the output function
- Digital
 - 1 switch on: Triggers when the digital output turns on
 - 0 switch off: Triggers when the digital output turns off

• **UDP**

Enter in the textfield 'CommandString' the specific string that you will send from your remote application.



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

- **Keystroke**

Enter the key character in the textfield

- On key press: Triggers when the key with the entered character is pressed
- On key release: Triggers when the key with the entered character is released

- **HTTP request**

Enter a string in the textfield. If zapy receives a http request it compares the string after the domain name/ip with the string entered. If this string matches this action will be triggered. (example: a http request to http://192.168.1.23/helloworld matches the entered string 'helloworld').

- **Get timed command**

Enter the command string in the textfield. This action will be triggered by a 'Call timed command' output event with the same command string. This can be used to enter a delay, execute a second timed command after an action or to continuously repeat an action.



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

7. Output actions in detail

There are different output actions zapy may execute. You may specify them in the I/O control panel.

• SWF function (for SWF-Display)

Enter a string into the textfield. zapy will call the function 'QRemoteCmd' in the root of the background swf and passes this string. The function in the background swf has to look something like this:

```
//-----  
function QRemoteCmd(getCmd: String) {  
    switch (getCmd){  
        case...  
    }  
}  
//-----
```

• UDP

Enter ip, port and command. The command then will be sent via UDP to the specific device using the entered ip and the specified port.

• Websocket

Enter a command string. This string will be sent to the remote application via websockets.

• Phidget

This event controls the phidgets attached. First select the phidget.

- InterfaceKit digital outputs:

- Switch On/Off: Changes output state of specified output.
- Turn On: Turns the specified output on.
- Turn Off: Turns the specified output off.
- Turn On for 200ms: Turns the specified output on for 0,2 secs then turns off again. May be used for coils and other short impulse.

- AdvancedServo Controller

- slowfade: Moves slowly to specific position
- fastfade: Moves fast to specific position
- setval: Moves as fast as possible to specific position



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

- **Zapy Event**

There is currently one zapy event for output actions:

- End websocket user session: Ends a user session

- **Analytics**

Enter a string which will be tracked by Google Analytics

- **Call timed command**

Enter a string and a delay in milliseconds. The string will be sent to the I/O events after the entered delay.

- **Javascript/HTML (for HTML-Display)**

Select one of the following options:

- load URL: loads the URL entered in the text field
- scroll down: scrolls page down
- scroll up: scrolls the page up
- javascript: executes the javascript code entered in the text field on the loaded page
- show QR Code: shows the page displaying the QR code



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

8. Remote application (HTML/CSS/JS)

The remote application is based on HTML, CSS and JS and may be easily configured according to your needs. Basically it's a simple webpage which will be displayed in the clients browser window. The remote application can be stored on the server of your choice. You may enter the URL/address in the general settings of the controll panel.

URL GUI-HTML:	<input type="text" value="http://www.zapy.ch/remotes/zapydefault/ws.html"/>	<input type="button" value="Save"/>
---------------	---	-------------------------------------

You need the following files:

- The HTML file which includes the form/buttons to controll zapy via websockets
- The CSS file which styles your application
- The JavaScript-file which connects to the websocket server and sends the commands
- The HTML file which will be displayed after the session end

To edit the remote application files you need a basic knowledge of HTML and CSS. The JS file doesn't need to be edited.

You may download a sample package at <http://www.zapy.ch/downloads/remoteapplication.zip>

Please make your adjustments in the HTML and CSS file. You don't have to modify the JS-file (except to modify some display texts).

On the following sites you will find a more detailed description of these files.



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

8.1. Remote application HTML file

In the remote application HTML file you may define the input elements.

```
<!doctype html>
<html class="html">
<!-- sets the width of the page to device-width-->
<meta name="viewport" content="width=device-width, user-scalable=no" />
<head>
<meta charset="UTF-8">
<title>ZapyRemote</title>
<!-- the css-file includes the style definitions -->
<link rel="stylesheet" type="text/css" href="zapy_remote.css" />
</head>
<body>
<!-- zapy messages will be printed into this div -->
<div class="zapy_msg" id="msg">---</div>
<!-- define the button class as 'zapy_ButtonDisabled'. The js will enable them ('zapy_Button') on websocket connection -->
<!-- on touch/click on the button call 'sendcmd(STRING)'. This string will be sent to the zapy application -->
<!-- Always use 'ontouchstart' and 'onmousedown' together to ensure that buttons will work on iOS, Android and Windows Phone -->
<div class="zapy_ButtonDisabled" ontouchstart = "sendcmd('triggerA')" onmousedown="sendcmd('triggerA')" >Button1</div>
<div class="zapy_ButtonDisabled" ontouchstart = "sendcmd('triggerB')" onmousedown="sendcmd('triggerB')" >Button2</div>
<!-- Include the zapy-javascript-file -->
<script type="text/javascript" src="zapy.js"></script>
</body>
</html>
```



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

8.2. Remote application CSS file

In the remote application CSS file you may define the styles.

These styles are used in the sample:

```
.html{...}
body {...}
.zapy_gui{...} – includes different commands to prevent application from zooming when touching a button
.zapy_msg{...} – the style for the message div
.zapy_Button, .zapy_ButtonDisabled {...} – the basic button styles
.zapy_Button {...} – style for active buttons
.zapy_Button .large {...} – large buttons will be displayed one below the other in portrait mode
.zapy_Button .small {...} – small buttons will be displayed in rows to three in landscape mode
.logo {...}
.imageclass {...}
.zapy_Button:active {...} – style for buttons pressed
.zapy_ButtonDisabled {...} – style for buttons disabled
```

8.3. Remote application JS file

The JS file doesn't need to be modified much. There are four variables which may be adjusted to your needs.

```
...
//*****
//
// zapy remote application
//
//*****

// Please modify the variables below to fit your needs
var msgYourTurn = "Steuern Sie mit den Tasten!"; // Will be displayed on successfull connection
var msgEnd = "Ihre Zeit ist um. Danke für Ihr Interesse."; // Will be displayed at session timeout
var msgConnectionRefused = "Verbindung fehlgeschlagen. Bitte scannen Sie den Code erneut."; // Displayed at connection failure
var zapy_endpoint = "zapy_end.html"; // The URL of the page which will be shown at session end

var counter_waiting = 0;
var counter_active = 0;
var waiting = false;
var active = false;
var msgdiv = document.getElementById("msg");
function disableGUI(){
    var elements = document.getElementsByClassName('zapy_Button');
    var i, len = elements.length;
    for (i = len-1; i >= 0; i--) {
        jsDisableElement(elements[i]);
    }
}
}
....
```



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

8.4. Remote application HTML exit file

The remote exit file is a simple HTML file which may contain a message/image/ad of your choice. You may link it to your onlineshop or everything you like.

A close-up, high-resolution photograph of a human eye, looking directly at the camera. The eye is brown with a dark pupil and a bright reflection. The surrounding skin and eyelashes are visible in detail. The lighting is warm and focused on the eye.

Be **surprised** by new ideas!



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch

9. License

Tryout: You may test zapy as long as you want and with all options included. The unlicensed version will shut down automatically after a few minutes and must be restarted manually.

License: A license may be ordered within the zapy application itself using the control panel tab ‚license‘ (see 5.6.). It is a ‚single computer license‘. This means that the license is only valid for the computer on which you filled out the license request. This procedure gives you the certainty that zapy will run on the computer you choosed for your application.

License costs:	Activation fee:	CHF 500.–	EURO 420.–	USD 520.–
	Monthly license:	CHF 30.– per month	EURO 26.– per month	USD 32.– per month
	Yearly license:	CHF 250.–	EURO 210.–	USD 260.–

Websocket server:	Development WSS*:	CHF 0.– per month	EURO 0.– per month	USD 0.– per month
	Shared WSS*:	CHF 30.– per month	EURO 26.– per month	USD 32.– per month
	Dedicated WSS*:	CHF 60.– per month	EURO 50.– per month	USD 62.– per month

*WSS = Websocket Server. WSS are provided by achex.ca. Prices may vary.

Non-commercial: There is no arrangement about non-commercial projects. Please contact us if you'd like to use zapy for a reasonable non-commercial project and we will check if you may get a discount.



Dissenter GmbH
Stadthausstrasse 65
8400 Winterthur

info@dissenter.ch
www.dissenter.ch