



TEC-IT

WWW.TEC-IT.COM

Scan-IT to Office

Add-In Developer Documentation

Version 1.0

30 November 2018

TEC-IT Datenverarbeitung GmbH
Hans-Wagner-Strasse 6
A-4400 Steyr, Austria

t ++43 (0)7252 72720
f ++43 (0)7252 72720 77
office@tec-it.com
www.tec-it.com

1 Content

1	Content	2
2	Disclaimer	3
3	Introduction	4
3.1	What is Scan-IT to Office?	4
3.2	Develop new Add-Ins	4
4	Protocol	5
4.1	General Message Format	5
4.1.1	Response Message Format	6
4.2	Add-In Identification	6
4.2.1	Add-In ID	6
4.2.2	Add-In Name	6
4.2.3	UPDATEADDINDATA Message	7
4.2.3.1	Response	7
4.3	Pairing	8
4.3.1	GETPAIRINGCODE Message	9
4.3.2	STARTPAIRING Message	9
4.3.3	STOPPAIRING	10
4.3.4	ID Response	10
4.3.5	SYNC Message	11
4.3.6	REMOVESYNC Message	12
4.4	Subscribe / Unsubscribe	13
4.4.1	SUBSCRIBE Message	13
4.4.1.1	Response	14
4.4.2	UNSUBSCRIBE Message	15
4.4.2.1	Response	16
4.5	Receiving Data from Devices	17
4.5.1	PUBLISH Message	17
4.5.1.1	Form Data Format	18
5	Server	19
5.1	Development Servers	19
5.2	Productive Server	19
6	Contact and Support Information	20

2 Disclaimer

The actual version of this product (document) is available as is. TEC-IT declines all warranties which go beyond applicable rights. The licensee (or reader) bears all risks that might take place during the use of the system (the documentation). TEC-IT and its contractual partners cannot be penalized for direct and indirect damages or losses (this includes non-restrictive, damages through loss of revenues, constriction in the exercise of business, loss of business information or any kind of commercial loss), which is caused by use or inability to use the product (documentation), although the possibility of such damage was pointed out by TEC-IT.



We reserve all rights to this document and the information contained therein. Reproduction, use or disclosure to third parties without express authority is strictly forbidden.



Für dieses Dokument und den darin dargestellten Gegenstand behalten wir uns alle Rechte vor. Vervielfältigung, Bekanntgabe an Dritte oder Verwendung außerhalb des vereinbarten Zweckes sind nicht gestattet.

© 1998-2018
TEC-IT Datenverarbeitung GmbH
Hans-Wagner-Str. 6

A-4400 Austria
t.: +43 (0)7252 72720
f.: +43 (0)7252 72720 77
www.tec-it.com

3 Introduction

3.1 What is Scan-IT to Office?

Scan-IT to Office is a solution for mobile data acquisition with real-time connectivity to suitable target applications (e.g. a spreadsheet application or a database).

With the **Scan-IT to Office** app (available for *Android* and *iOS*), users can capture data with a mobile device and send them (via a *WebSocket* server) to the connected target application(s).

The **Scan-IT to Office** Add-In receives the sent data and inserts it accordingly.

3.2 Develop new Add-Ins

Scan-IT to Office was developed with extensibility in mind.

If there is no **Scan-IT to Office** Add-in available for your desired target application, you can develop a new Add-In by following this documentation.

The communication between our server and the Add-Ins is following a well-defined protocol.

JSON formatted, secure *WebSocket* messages are used to request data from the server(s) (e.g. a pairing code), inform the server about updated Add-In data (e.g. Add-In ID and name) and to subscribe to or unsubscribe from a devices channel (publish/subscribe system).

On the following pages you can find a detailed documentation of this protocol.

4 Protocol

4.1 General Message Format

A message from the Add-In to the server will usually look like this:

```
{
  "type" : "<MESSAGETYPE>",
  ...
  "addin" :
  {
    "addinId" : "<GUID>",
    "name" : "<NAME>"
  },
  "lang" : "<LOCALE>",
  "versionCode" : "<VERSIONCODE>"
}
```

- **<MESSAGETYPE>**:
Message type identifier.
- **<GUID>**:
See below (4.2.1 Add-In ID)
- **<NAME>**:
See below (4.2.2 Add-In Name)
- **<LOCALE>**:
The users locale. E.g. "en-US". Server will currently ignore this but may send response messages accordingly in the future.
- **<VERSIONCODE>**:
The Add-In type and the version of the Add-In in the following format:
"<TYPEIDENTIFIER>_<VERSION>"
A Microsoft Access Add-In for example could have the <TYPEIDENTIFIER> "MA", the <VERSIONCODE> would then look like this: "MA_1.0.0". Please clarify with us which <TYPEIDENTIFIER> to use.

4.1.1 Response Message Format

A typical response message from the server to the Add-In will usually look like this:

```
{
  "type" : "<RESPONSETYPE>",
  "code" : "<RESPONSECODE>",
  "data" : "<MESSAGETEXT>"
  ...
}
```

- <RESPONSETYPE>:
Response type identifier. Can be "INFO", "ERROR" or "ID".
- <RESPONSECODE>:
Currently this is always *null*, in the near future we will add codes to identify the possible response messages more easily.
- <MESSAGETEXT>:
The message text. E.g. "Registered device <DEVICEID>".

4.2 Add-In Identification

4.2.1 Add-In ID

In order for the Add-In to be identified by the server, it is necessary to generate and send a *GUID*.

The Add-In needs to persist this *GUID* locally and send it with every message.

4.2.2 Add-In Name

It is possible for the user to select a name for the Add-In, so that it can be identified in the app.

The default name should contain the name of the target application (eg. "Microsoft Access Add-In").

The Add-In needs to persist the name locally and send it with every message.

When the name of an Add-In is changed by the user, inform the server about this change with an *UPDATEADDINDATA* message.

4.2.3 UPDATEADDINDATA Message

When the user changes the name of an Add-In, you need to inform the server about that change with an *UPDATEADDINDATA* message.

Message Format:

```
{
  "type" : "UPDATEADDINDATA",
  "addin" :
  {
    "addinId" : "<GUID>",
    "name" : "<NAME>"
  },
  "lang" : "<LOCALE>",
  "versionCode" : "<VERSIONCODE>"
}
```

- <GUID>: The *GUID* generated by the Add-In.
- <NAME>: The name of the Add-In.

4.2.3.1 Response

Message Format:

```
{
  "type" : "<RESPONSETYPE>",
  "code" : null,
  "data" : "<MESSAGETEXT>"
}
```

- <RESPONSETYPE>:
INFO or ERROR.
- <MESSAGETEXT>:
Possible message texts are:
 - "Add-In data updated" (INFO)
 - "Operation failed. Parameter addin.addinId must be set" (ERROR)

4.3 Pairing

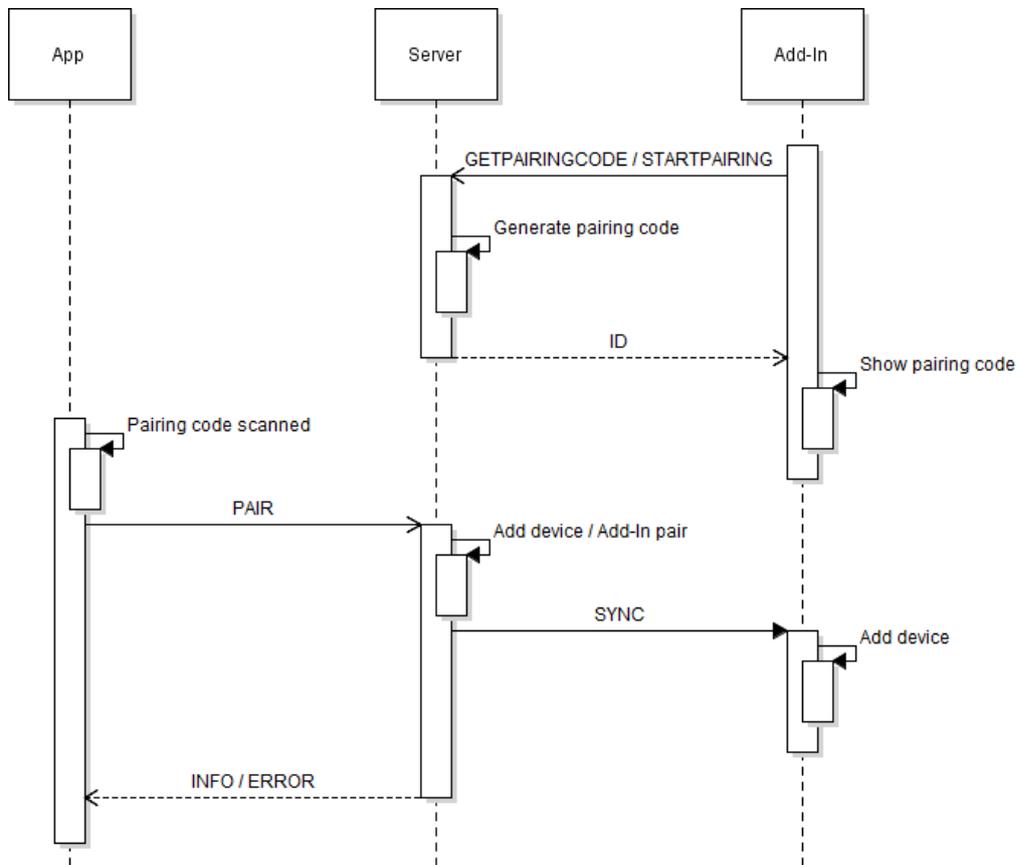


Figure 1: Pairing lifecycle

To start a pairing process you need to request a pairing code from the server by sending a `GETPAIRINGCODE` or a `STARTPAIRING` message.

With the `GETPAIRINGCODE` message, the server will only generate one pairing code that will be valid for 10 minutes.

With the `STARTPAIRING` message, the server will continuously generate pairing codes every 10 minutes. You need to send a `STOPPAIRING` request when your pairing process is finished or cancelled.

Since our server is designed to store minimal personal user data, you have to make sure to persist all connected devices locally, so you are able to register them on the server again when the Add-In is started the next time.

4.3.1 GETPAIRINGCODE Message

To request a single pairing code (valid for 10 minutes), send a *GETPAIRINGCODE* message to the server. The server will respond with an "ID" response once (see 4.3.4 ID Response).

Message Format:

```
{
  "type" : "GETPAIRINGCODE",
  "addin" :
  {
    "addinId" : "<GUID>",
    "name" : "<NAME>"
  },
  "lang" : "<LOCALE>",
  "versionCode" : "<VERSIONCODE>"
}
```

4.3.2 STARTPAIRING Message

To request pairing codes continuously, send a *STARTPAIRING* message to the server. The server will respond every 10 minutes with an "ID" response (see 4.3.4 ID Response).

Message Format:

```
{
  "type" : "STARTPAIRING",
  "addin" :
  {
    "addinId" : "<GUID>",
    "name" : "<NAME>"
  },
  "lang" : "<LOCALE>",
  "versionCode" : "<VERSIONCODE>"
}
```

4.3.3 STOPPAIRING

When you requested continuous generation of pairing codes with a *STARTPAIRING* message, you can stop the generation of new pairing codes with a *STOPPAIRING* message.

Message Format:

```
{
  "type" : "STOPPAIRING",
  "addin" :
  {
    "addinId" : "<GUID>",
    "name" : "<NAME>"
  },
  "lang" : "<LOCALE>",
  "versionCode" : "<VERSIONCODE>"
}
```

4.3.4 ID Response

When a pairing code was requested the server will respond with an *ID* response.

Message Format:

```
{
  "type" : "ID",
  "code" : null,
  "data" : "<PAIRINGCODE>",
  "timeout" : "<TIMEOUT>",
}
```

- **<PAIRINGCODE>:**
A hexadecimal string with 8 digits length. E.g. "A0B1C2D3". Display as QR code and as plain text so users can pair by scanning the QR code with the app or by entering the code manually.
You can use third party libraries or online APIs to generate the QR code, for example <https://qrcode.tec-it.com/API/QRCode?data=<PairingCode>>
- **<TIMEOUT>:**
Timeout in seconds. You can use this to display a progress bar or countdown. If you used a *STARTPAIRING* message to request the pairing code you will automatically get a new *ID* response with a pairing code after this time. If you used a *GETPAIRINGCODE* message, you will get a new *ID* response with *null* data after that time.

4.3.5 SYNC Message

When a device connects to your Add-In by scanning or entering the displayed pairing code, the server will send a *SYNC* message with device data to the corresponding Add-In.

Message Format:

```
{
  "type" : "SYNC",
  "device":
  {
    "deviceId" : "<DEVICEID>",
    "password" : "<PASSWORD>",
    "displayName" : "<DEVICENAME>"
  }
}
```

- <DEVICEID>:
A hexadecimal string with 16 digits length for device identification. E.g. "A0B1C2D3A0B1C2D3".
- <PASSWORD>:
A hexadecimal string with 16 digits length for device authorization. E.g. "A0B1C2D3A0B1C2D3".
- <DEVICENAME>:
The human readable name of the device. Display this name in your Add-In device list.

4.3.6 REMOVESYNC Message

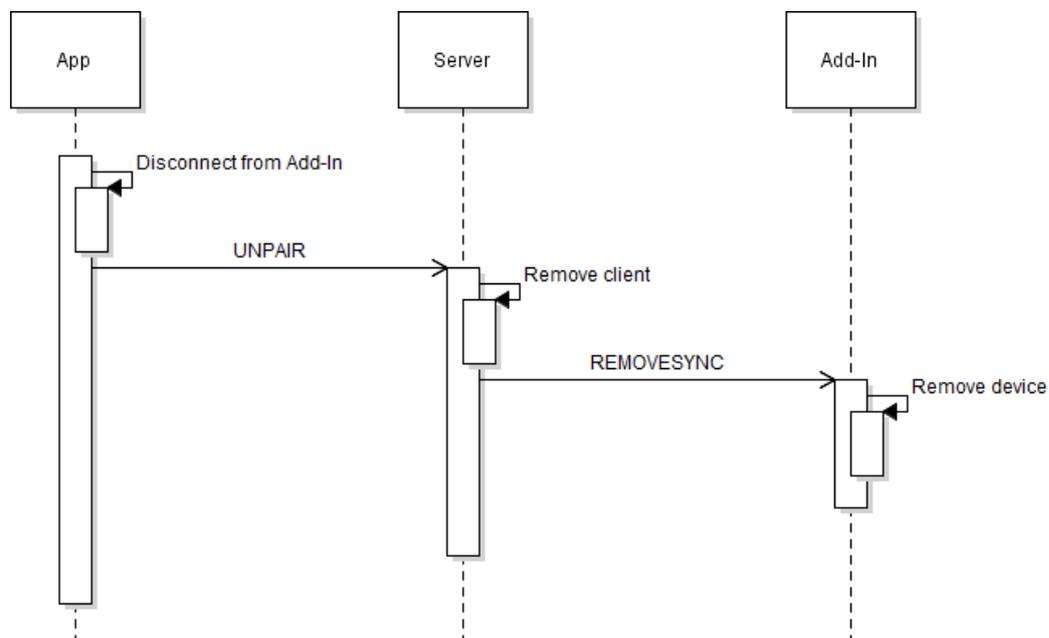


Figure 2: Disconnect lifecycle (device disconnects from Add-In)

When a device disconnects from an Add-In, the server will send a *REMOVESYNC* message to the corresponding Add-In.

If the Add-In is currently offline, the server will send the *REMOVESYNC* message when you try to subscribe again for the device (see 4.4 Subscribe).

Message Format:

```
{
  "type" : "REMOVESYNC",
  "device" :
  {
    "deviceId" : "<DEVICEID>",
    "password" : "<PASSWORD>",
    "displayName" : "<DEVICENAME>"
  }
}
```

4.4 Subscribe / Unsubscribe

4.4.1 SUBSCRIBE Message

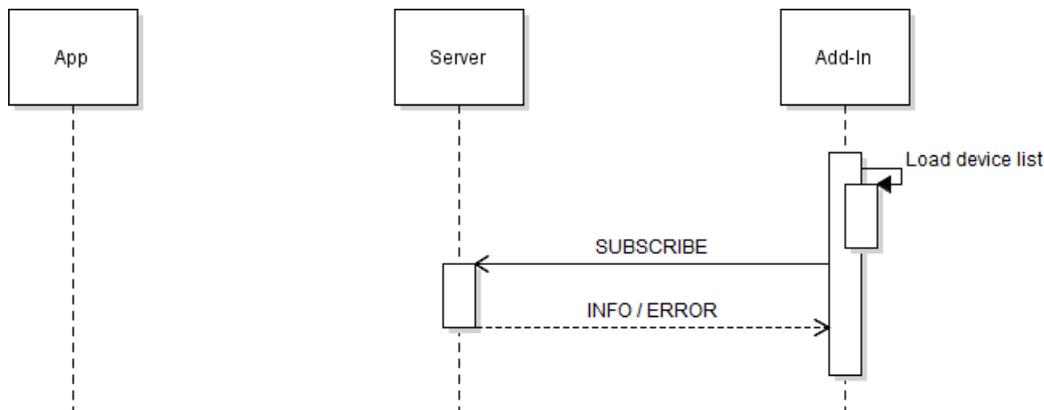


Figure 3: Subscribe lifecycle

Since our server is designed to store minimal personal user data, you have to make sure to persist all devices received within a *SYNC* message locally.

Whenever your Add-In establishes a new connection to our server (e.g. when the Add-In is started or when the connection was lost and reestablished), you have to subscribe to the devices you persisted before with a *SUBSCRIBE* message.

Message Format:

```

{
  "type" : "SUBSCRIBE",
  "addin" :
  {
    "addinId" : "<GUID>",
    "name" : "<NAME>"
  },
  "device" :
  {
    "deviceId" : "<DEVICEID>",
    "password" : "<PASSWORD>"
  },
  "id" : "<DEVICEID_UI>",
  "lang" : "<LOCALE>",
  "versionCode" : "<VERSIONCODE>"
}
  
```

- <DEVICEID_UI> (optional): Identifier that is returned with a response, so you can associate the ERROR or INFO response to a device in your user interface.

4.4.1.1 Response

Message Format:

```
{
  "type" : "<RESPONSETYPE>",
  "code" : null,
  "data" : "<MESSAGETEXT>"
  "id" : "<DEVICEID_UI>"
}
```

- <RESPONSETYPE>:
INFO or ERROR.
- <MESSAGETEXT>:
Possible message texts are:
 - "Registered device <DEVICEID>" (INFO)
The Add-In successfully subscribed for the device.
 - "Device disconnected from Addin <DEVICEID>" (INFO)
This means that the device disconnected from the Add-In while it was offline. The server will also send a REMOVESYNC message in this case (see 4.3.6 REMOVESYNC message).
 - "Device already registered <DEVICEID>" (ERROR)
You tried to subscribe to the same device multiple times.
 - "Operation failed. Invalid credentials" (ERROR)
The password of the device you sent was incorrect.
- <DEVICEID_UI>:
The identifier you sent with the SUBSCRIBE message.



4.4.2 UNSUBSCRIBE Message

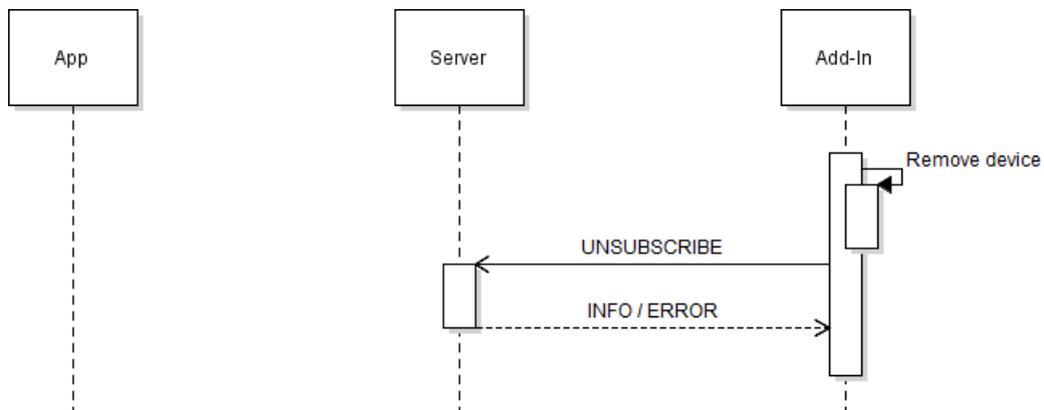


Figure 4: Unsubscribe lifecycle

When the user removes a device in the Add-In, you need to send an *UNSUBSCRIBE* message to the server.

Message Format:

```
{
  "type" : "UNSUBSCRIBE",
  "addin" :
  {
    "addinId" : "<GUID>",
    "name" : "<NAME>"
  },
  "device" :
  {
    "deviceId" : "<DEVICEID>",
    "password" : "<PASSWORD>"
  },
  "id" : "<DEVICEID_UI>",
  "lang" : "<LOCALE>",
  "versionCode" : "<VERSIONCODE>"
}
```

4.4.2.1 Response

Message Format:

```
{
  "type" : "<RESPONSETYPE>",
  "code" : null,
  "data" : "<MESSAGETEXT>"
  "id" : "<DEVICEID_UI>"
}
```

- <RESPONSETYPE>:
INFO or ERROR.
- <MESSAGETEXT>:
Possible message texts are:
 - “Unregistered device <DEVICEID>” (INFO)
The Add-In successfully unsubscribed from the device.
 - “Operation failed. Device not found <DEVICEID>” (ERROR)
The Add-In could not be unsubscribed from the device because it was not subscribed to it. You probably tried to unsubscribe multiple times or you didn’t subscribe successfully before.
 - “Operation failed. Invalid credentials” (ERROR)
The password of the device you sent was incorrect.
- <DEVICEID_UI>:
The identifier you sent with the SUBSCRIBE message.



4.5 Receiving Data from Devices

4.5.1 PUBLISH Message

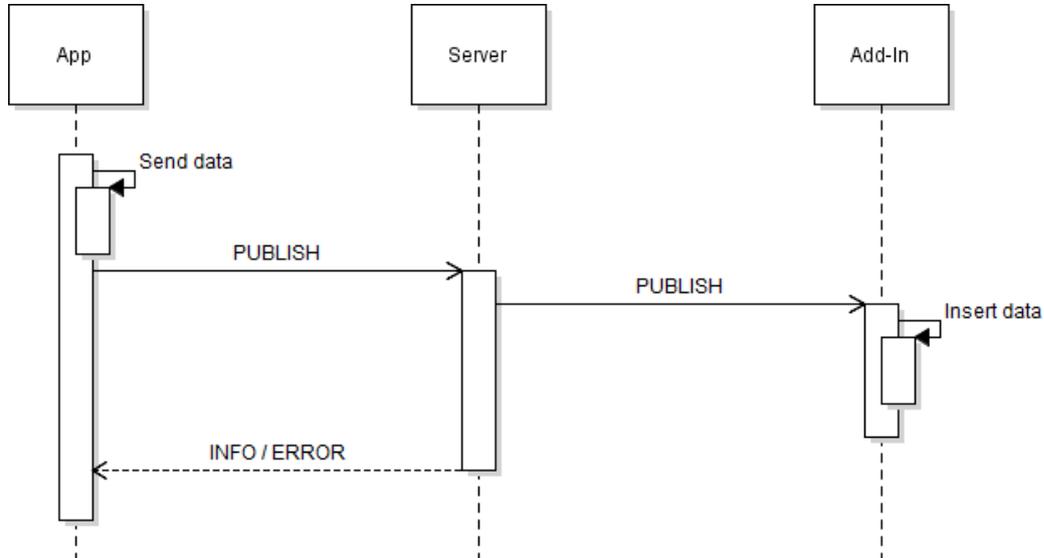


Figure 5: Publish lifecycle

Whenever a device is sending data, the server will send a *PUBLISH* message to all Add-Ins that are subscribed to the corresponding device.

Message Format:

```

{
  "type" : "PUBLISH",
  "data" : <DATA>,
  "device" :
  {
    "deviceId" : "<DEVICEID>",
    "password" : "<PASSWORD>",
    "displayName" : "<DEVICENAME>"
  },
  "timestamp" : "<ACQUISITIONTIMESTAMP>",
  "lang" : "<LOCALE>",
  "versionCode" : "<VERSIONCODE>"
}
  
```

- **<DATA>:**
If the user is just scanning single barcodes, this will be a plain string.
If the user is using a form, this will be a JSON Array with type information and data of all fields in that form (see 4.5.1.1 Form Data Format).
- **<ACQUISITIONTIMESTAMP>:**
A string in ISO 8601 notation representing the date and time when the data was captured. It will contain the local time zone of the device.
E.g. "2018-11-30T13:37:00+01:00"
If you want to display this data, you should consider converting it to the local time zone of the receiving Add-In.

4.5.1.1 Form Data Format

When a user is sending a form, the *PUBLISH* messages “data” property will contain a JSON array with type information and the values of all form fields.

Message Format:

```
{
  "type" : "PUBLISH",
  "data" :
  [
    {
      "type" : "<FIELDTYPE>",
      "value" : "<VALUE>"
    }, ...
  ],
  "device" :
  {
    "deviceId" : "<DEVICEID>",
    "password" : "<PASSWORD>",
    "displayName" : "<DEVICENAME>"
  },
  "timestamp" : "<ACQUISITIONTIMESTAMP>",
  "lang" : "<LOCALE>",
  "versionCode" : "<VERSIONCODE>"
}
```

- **<FIELDTYPE>:**
Possible types are:
 - “text/plain” → (barcode, text, number, list, checkbox, location and static text fields)
 - “text/x.date” → (date fields)
 - “text/x.time” → (time fields)
 - “text/x.datetime” → (date & time fields)
 - “image” or “image/jpeg” → (image fields)
- **<VALUE>:**
 - “text/plain” → A string that can contain control characters
 - “text/x.date” → A string in ISO 8601 notation. E.g. “2018-11-30T13:37:00Z”
 - “text/x.time” → A string in ISO 8601 notation. E.g. “2018-11-30T13:37:00Z”
 - “text/x.datetime” → A string in ISO 8601 notation. E.g. “2018-11-30T13:37:00Z”
 - “image” or “image/jpeg” → A base64 String with image data.

- ▶ Date, time and date & time fields are converted to UTC by the app. Convert them to the local time zone of the Add-In.
- ▶ If your Add-In does not support images, you should display an appropriate hint.

5 Server

5.1 Development Servers

Please use one of our development servers in your development process.

- `wss://sto-dev.azurewebsites.net:443`
- `wss://stodev.azurewebsites.net:443`

- ▶ Note: These servers can sometimes be outdated or inactive, please contact us when your development process starts and we will activate / update the servers.
- ▶ To change the server in the app, a subscription is needed. Contact us so we can provide access to subscribed versions of the app free of charge during your development process.

5.2 Productive Server

When your Add-In is finished and tested, you can switch to our productive server:

- `wss://tecit-sto-demo00.azurewebsites.net:443`

- ▶ Please do not use the productive server for development!

6 Contact and Support Information

TEC-IT Datenverarbeitung GmbH

Address: Hans-Wagner-Str. 6
AT-4400 Steyr
Austria/Europe
Phone: +43 / (0)7252 / 72 72 0
Fax: +43 / (0)7252 / 72 72 0 – 77
Email: office@tec-it.com
Web: www.tec-it.com

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C, World Wide Web Consortium, Laboratory for Computer Science NE43-358, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139.

JAVASCRIPT® is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

Microsoft®, Windows®, Microsoft Word®, Microsoft Excel® and Microsoft Access® are registered trademarks of Microsoft Corporation.

Apple®, App Store®, Mac®, Mac OS®, macOS®, OS X®, iPhone®, iPad® and iOS® are registered trademarks of Apple Inc., registered in the U.S. and other countries.

Android™, Google Play™, Google Docs™ and Google Sheets™ are trademarks of Google Inc.

All other products mentioned are trademarks or registered trademarks of their respective companies. If any trademark on our web site or in this document is not marked as trademark (or registered trademark), we ask you to send us a short message (office@tec-it.com).

