Event Specification

1 Requirements

The connecting portal client has requested a signed certificate from the portal server and the signed certificate is available locally

2 API endpoints

The API for pushing events to the portal server will be available via HTTPS and websocket over HTTPS.

The portal client must supply its certificate to the portal server when establishing a connection. If the portal client doesn't send its certificate to the portal server or the certificate is invalid (revoked, not signed by the ISPf Root CA), the portal server will respond with the HTTP status code 403 "Forbidden".

3 Event format

3.1.1 Data format

The data format for events will be JSON.

3.1.2 Required information in event format

3.1.2.1 ID

The portal client must generate an id for every event. The id must be an UUID (http://en.wikipedia.org/wiki/Universally_unique_identifier) (Version 4).

3.1.2.2 Timestamp

The portal client must supply a local timestamp for the event. The timestamp must be compatible to the ISO 8601 format (http://en.wikipedia.org/wiki/ISO_8601).

3.1.2.3 Type

The portal client must supply the type of the event. See document „Event types“ for a list of available events.

3.1.3 Optional information in event format

3.1.3.1 Belongs-To

The portal client can supply an optional reference to another event within „belongsto“.

3.1.3.2 Payload

The portal client can supply an optional „payload“ for the event (textual information, JSON, XML, XML-RPC, images, ...). Given, that the chosen data format, JSON, doesn’t support multiline strings for values, the payload must be encoded with Base64.

3.1.3.3 Destination

The portal client can supply an array of recipient-UUIDs for the event. See chapter „Discovery“ of document „Event types“.

If no value for an optional field is present, the optional field can be filled with a *null*–value or omitted completely.

3.1.4 Example

See data/example/event.json

```
{

   "id": "550e8400-e29b-41d4-a716-446655440000",

   "timestamp": "2008-02-01T09:00:22+05:00",

   "type": "com.abb.ispf.event.echo",

   "belongsto": "550e8400-e29b-41d4-a716-446655440000",

   "payload": "BASE64 ENCODED STRING CONTAINING PAYLOAD DATA",

   "destination": []

}
```

3.2 Connection via Websocket over HTTPS

3.2.1 Websocket API endpoint

The websocket portal server will be available under the following URL path:

/socket

The complete URL for the websocket API endpoint will be

- wss://staging.ispf.desto.datadevelopment.de/socket
- or respectively wss://staging.ispf.debje.datadevelopment.de/socket

3.2.1.1 Processing order of an incoming websocket message

- The portal server will parse the incoming JSON messages
- The portal server will validate the supplied event type
- The portal server will verify if the supplied id is unique in its database
- The portal server will verify if the supplied belongsto id is available in its database

3.2.1.2 Acknowledgement for incoming websocket messages

3.2.1.2.1 In case of successful processing by the portal server

If the validation of the incoming message succeeded, the portal server will send a message as acknowledgement back to the portal client:

```
{

   "id": "7C65AA03-8AD7-4213-94A7-39B0CAB8714A",

   "timestamp": "2008-02-01T09:00:22+05:00",

   "type": "com.abb.ispf.event.success",

   "belongsto": "ID_OF_MESSAGE_FROM_PORTAL_CLIENT"

}
```

3.2.1.2.2 In case of an error

If the validation of the incoming message did not succeed, the portal server will send a message back to the portal client:

```
{
  "id": "C7E01BF5-7760-4E7A-B61A-7519EB7F3097",
  "timestamp": "2008-02-01T09:00:22+05:00",
  "type": "com.abb.ispf.event.error",
  "belongsto": "ID_OF_MESSAGE_FROM_PORTAL_CLIENT",
  "payload": "BASE64 ENCODED STRING WITH DETAILS ABOUT WHAT WENT WRONG"
}
```

The decoded Base64 payload data of the error event could have the following formings:

```
{
  "code": 409,
  "message": "Supplied id is not unique to the portal server"
}
{
  "code": 400,
  "message": "Supplied belongsto id is unknown to the portal server"
}
{
  "code": 400,
  "message": "Supplied type is unknown to the portal server"
}
```

3.3 Connection via HTTPS

3.3.1 HTTP API endpoint

The API for pushing events to the portal server will be available under the following URL path:

/api/event

3.3.1.1 The complete URL for the HTTP API endpoint will be

- https://staging.ispf.desto.datadevelopment.de/api/event
- or respectively https://staging.ispf.debje.datadevelopment.de/api/event

3.3.1.2 Accepted HTTP methods

- GET (request body will be ignored; will be specified later; used to query the portal server for events by criterias)
- POST (request body will be parsed; used to push new event to portal server)

If the portal client sends a request with a different HTTP method than the ones mentioned above, the portal server will respond with the HTTP status code 405 „Method not allowed“.

3.3.1.3 Pushing an event to the HTTP API endpoint

In order to push a new event to the portal server, the portal client must use the HTTP method „POST“. The portal client must send the generated JSON data as request body to the API. If the validation of the sent request body fails, the portal server will respond with the HTTP status code 400 "Bad Request". If the provided event id is not unique to the portal server, the portal server responds with the HTTP status code 409 "Conflict". The event must be send again with a new id.

If all went well, the portal server responds with the HTTP status code 201 "Created".

3.3.1.3.1 Summary of possible HTTP status codes when pushing to server

- 201 - Created; Everything went well
- 400 - Bad request; request body was malformed
- 403 - Forbidden; Certificate was not supplied to the portal server, certificate was revoked or not signed by the ISPf Root CA
- 405 - Method not allowed; Wrong HTTP method was used; only GET and PUT are allowed
- 409 - Conflict; an event with this id was already stored on the portal server
- 500 - Something along the processing on the portal server went wrong

3.3.1.3.1.1 Reasons for the returned HTTP status code 400 "Bad Request"

- Provided request body cannot be parsed (invalid JSON, …)
- Provided event type is unknown
- Provided belongsto id is unknown to the portal server

3.3.1.4 Fetching events from the HTTP API endpoint

3.3.1.4.1 Fetching a single event from the HTTP API endpoint

In order to fetch a single event from the portal server, the portal client must use the HTTP method „GET“. The ID of the requested event must be added to the HTTP API endpoint URL.

3.3.1.4.1.1 Example

/api/event/550e8400-e29b-41d4-a716-446655440000

3.3.1.4.1.2 Result

```
{

  "id": "550e8400-e29b-41d4-a716-446655440000",

  "timestamp": "2008-02-01T09:00:22+05:00",

  "timestamp_portal": "2008-02-01T09:00:22+05:00",

  "type": "com.abb.ispf.event.echo",

  "belongsto": "550e8400-e29b-41d4-a716-446655440000",

  "payload": "BASE64 ENCODED STRING CONTAINING PAYLOAD DATA",

  "destination": null

}
```

3.3.1.4.2 Fetching multiple events from the HTTP API endpoint

In order to fetch multiple events from the portal server, the portal client must use the HTTP method „GET“.

3.3.1.4.2.1 The API will support the following query parameters:

3.3.1.4.2.1.1 Type of event – type

Fetch only events of a specific type. A comma-separated list of different event types will also be supported.

type=com.abb.ispf.event.echo,com.abb.ispf.event.deviceinfo

3.3.1.4.2.1.2 ID of an event – id

Fetch only events with a specific ID. A comma-separated list of IDs will also be supported.

id=550e8400-e29b-41d4-a716-446655440000,C7E01BF5-7760-4E7A-B61A-7519EB7F3097

3.3.1.4.2.1.3 Newer than ID – newer_than_id

Fetch only events that are newer than the one with the ID specified.

newer_than_id=0A51AD34-0D8E-4835-915E-6AC7A984F331

3.3.1.4.2.1.4 Newer than point in time – newer_than

Fetch only events, that occured after the specified timestamp. The specified timestamp must be in ISO 8601 format.

newer_than=2008-02-01T09:00:22+05:00

3.3.1.4.2.2 Older than ID – older_than_id

Fetch only events that are older than the one with the ID specified.

older_than_id=6A87F1BE-C825-4931-904C-34CCBE00B693

3.3.1.4.2.2.1 Older than point in time – older_than

Fetch only events that are older than the specified timestamp. The specified timestamp must be in ISO 8601 format.

older_than=2013-09-10T16:00:22+01:00

3.3.1.4.2.2.2 Belonging to ID – belongsto

Fetch only events that are belonging to the event ID specified. A comma-separated list of IDs will also be supported.

belongsto=A2715FEF-7CBA-4A32-973C-DD25A4FC456E,8956AAD2-E231-4B74-AE6D-B091EF37221E

3.3.1.4.2.2.3 Portal client – portal_client

Fetch only events that were send from the specified portal client (UUID of portal client). A comma-separated list of portal clients will also be supported.

portal_client=BF815D03-2619-4F10-AED1-58BC8FE6AB26,EEC490E4-0D88-4F41-B6E1-6BEFD9A273C9,0E574ADB-8E39-49DC-A825-C0780351EF49

3.3.1.4.2.3 Pagination

Under some circumstances, the list of matching events can be very long. Therefore, the API will provide the possibility, to fetch only a smaller set of matching events.

The following query parameters can be used, to split the matching events into pages:

Number of returned events per page – pagination_limit (positive, non-zero integer)

Current page number – pagination_page (positive, non-zero integer)

3.3.1.4.2.4 Example

/api/event/?type=com.abb.ispf.event.echo&newer_than=2013-08-31T00:00:00+01:00&pagination_limit=3&pagination_page=2

3.3.1.4.2.5 Result

```
{
  "count_total": 10,
  "events": [
    {
      "id": "DFBE0010-327E-49D4-BD3C-7D64A8D0143A",
      "timestamp": "2013-09-10T08:12:22+01:00",
      "timestamp_portal": "2013-09-10T08:12:22+01:00",
      "type": "com.abb.ispf.event.echo",
      "belongsto": null,
      "payload": "SGVsbG8gV29ybGQ=",
      "destination": null
    },
    {
      "id": "FA16130F-CA78-41A4-BB87-71123A4E5695",
      "timestamp": "2013-09-10T07:44:09+01:00",
      "timestamp_portal": "2013-09-10T07:44:09+01:00",
      "type": "com.abb.ispf.event.echo",
      "belongsto": null,
      "payload": "Zm9v",
      "destination": null
    },
    {
      "id": "7242410E-68FF-4B4E-8292-F7C8C56821F7",
      "timestamp": "2013-09-09T13:54:17+01:00",
      "timestamp_portal": "2013-09-09T13:54:17+01:00",
      "type": "com.abb.ispf.event.echo",
      "belongsto": null,
      "payload": "YmFy",
        "destination": null
      }
   ]
}
```

In this example, we requested the API to:

- Only return events of type „com.abb.ispf.event.echo“
- Only return events that are newer than 31th August 2013
- Only return three events per page
- Return the second page of matching events

The response body is supplemented by the total count of matching events. The matching events are embedded in the key „events“.

4 Event Type Definitions

4.1 General Event Types

IS-861 - Echo event – com.abb.ispf.event.echo

Direction	Bi-Directional
Purpose	Used to test the websocket connection.
Usage	The websocket server of the portal will send a message back to the portal client, containing the payload data, the portal client sent to the portal server.
Status	Open

IS-865 - Success – com.abb.ispf.event.success

Direction	Server to Client
Purpose	Used to acknowledge the reception of an event from a websocket portal client.
Usage	
Status	Open

IS-866 - Error – com.abb.ispf.event.error

Direction	Server to Client
Purpose	Used to inform a portal client about an error that occured while processing the last portal clients message.
Usage	
Status	Open

IS-867 - Device info – com.abb.ispf.event.deviceinfo

Direction	Client to Server.
Purpose	Used to inform the portal server about: • Hardware version (not applicable for apps) • Software version • Current system time • Feature set (only applicable for apps, e.g. “Can the camera be used?”)

Usage	Example of payload:
	{
	"hardware-version": "1.0",
	"software-version": "1.0",
	"timestamp": "2008-02-01T09:00:22+05:00",
	"feature-set": {
	"camera": true
	}
	}
	Further information about the feature-set should be provided by DEBJE.

Status	Open

IS-868 - Discovery – com.abb.ispf.event.discovery

Direction	Server to Client
Purpose	Whenever the user registers a new portal client, a discovery event will be generated by the portal server. This mechanism will be used to inform portal clients about their contactable counterparts (example: Busch-Welcome will be informed about the contactable Busch-Welcome IP-Gateways, KNX-Security-Panel-App will be informed about the contactable KNX-Security-Panel IP-Gateways).
Usage	The payload of the event will contain a list of contactable clients. This list will also contain the friendly name and type of the addressable portal clients. Example of the payload: { "84E7DFF7-44D0-48C2-80A9-99DB7C27CD86": { "name": "IP-Gateway, Loft", "type": "com.abb.ispf.client.welcome.gateway" }, "ADFB584B-BB24-46EA-A1A3-D39AE1A5F785": { "name": "IP-Gateway, Summer cottage", "type": "com.abb.ispf.client.welcome.gateway" },

	"AAF7B4BB-AECA-4420-A875-893947FEDD22": { "name": null, "type": "com.abb.ispf.client.welcome.gateway" }
	}

Status	Open

IS-869 - Delete event – com.abb.ispf.event.delete

Direction	Client to Server
Purpose	Used to inform the portal server about an event that should be deleted.

Usage	Example of event message:
	{
	"id": "7C65AA03-8AD7-4213-94A7-39B0CAB8714A",
	"timestamp": "2008-02-01T09:00:22+05:00",
	"type": "com.abb.ispf.event.delete",
	"belongsto": "ADFB584B-BB24-46EA-A1A3-D39AE1A5F785"
	}
	To delete a specific event, the UUID of the event must be put into the field „be-longsto“.

Status	Open

TO BE DISCUSSED:
We should not rely on the possibility to force the portal client to stop connecting to the portal by sending him a message (via HTTP-Long-Polling, Push-Notfications, Websocket, ...). The app may be closed, or the push notifications may be tempora-rily disabled. In this case, the information of a to be disabled connection may be lost.
To (temporarily) disable the remote access of a portal client, the certificate of the portal client should be added to the revocation list of the portal. Additionally, the SIP-server should provide the portal with an interface, where the updated revocati-on list can be pushed to. From our perspective, this is the only secure way to handle to prevent further connections from a stolen mobile device.

IS-870 - Delete client certificate – com.abb.ispf.event.client-remove

Direction	Server to Client
Purpose	Used to request the portal client to delete its certificate and to no longer connect to the portal.
Usage	When the portal client receives a event of this type, the client must visit the HTTP API endpoint for certificate revocation, which is further specified in document „Certificate revocation“.
Status	Open

4.2 Event types specific for KNX Security Panel

IS-872 - Alert – com.abb.ispf.event.knx-security-panel.alert

Direction	Client to Server
Purpose	Used to inform portal and apps about active alarms/faults in the KNX Security Panel.
Usage	

Status	Open

IS-862 - Command – com.abb.ispf.event.knx-security-panel.command

Direction	Bi-Directional
Purpose	From KNX Security Panel apps to portal server: Used to send a command request to the portal server (setting, unsetting, reset, ...). From portal server to KNX Security Panel: Used to transmit command request to the KNX Security Panel.
Usage	The KNX Security Panel app must specify the destination of the event (UUID of the KNX Security Panel portal client).
Status	Open

IS-873 - State – com.abb.ispf.event.knx-security-panel.state

Direction	Client to Server
Purpose	From KNX Security Panel: Used to send informations about the current state of the KNX Security Panel to the portal. The portal will store this information for later transmission to KNX Security Panel apps. From KNX Security Panel apps: Used to retrieve informations about the current state of the KNX Security Panel from the portal.
Usage	
Status	Open

IS-864 - Log – com.abb.ispf.event.knx-security-panel.log

Direction	Client to Server
Purpose	From KNX Security Panel: Used to send event logs of the KNX Security Panel to the portal server. The portal server will store this information for later transmission to the KNX Security Panel apps. From KNX Security Panel apps: Used to retrieve log entries of the KNX Security Panel.
Usage	
Status	Open

4.3 Event types specific for Busch-Welcome

IS-875 - Pairing between App and Gateway – com.abb.ispf.event.welcome.connect

Direction	Client (App) to Server / Client (Gateway)
Purpose	Used to send a request for authorization from App to an IP-Gateway.

Usage	The Busch-Welcome app must specifiy the destination IP-Gateway within the “desti-nation” field of the event format.
	Example for a complete event message:
	{
	"id": "7C65AA03-8AD7-4213-94A7-39B0CAB8714A",
	"timestamp": "2008-02-01T09:00:22+05:00",
	"type": "com.abb.ispf.event.welcome.connect",
	"destination": ["ADFB584B-BB24-46EA-A1A3-D39AE1A5F785"]
	}

Status	Open

IS-876 - ACL-Request – com.abb.ispf.event.welcome.acl-request

Direction	Server to Client (Gateway)
Purpose	Used to inform the Busch-Welcome IP-Gateway about an app, that want’s to get access.

Usage	Example of the payload data:
	{
	"uuid": "193CC63E-B21C-43F5-981D-D5BB32CFE3AA",
	"serial": "3127491",
	"name": "John Doe's iPhone"
	}

Status	Open

IS-877 - ACL-Upload – com.abb.ispf.event.welcome.acl-upload

Direction	Client (Gateway) to Server
Purpose	Used to inform the portal server about changes in the access control list of the IP-Gateway.
Usage	
Status	Open

IS-878 - ACL-Update – com.abb.ispf.event.welcome.acl-update

Direction	Server to Client
Purpose	This event is used to inform a portal client (Welcome-App) about granted privileges on one specific IP-Gateway.
Usage	

Status	Open

IS-879 - Surveillance call – com.abb.ispf.event.welcome.call-surveillance

Direction	Bi-Directional
Purpose	Portal client (Busch-Welcome IP-Gateway) to portal server. Used to inform the portal server about a started surveillance call. The event must include which client started the surveillance call. Portal server to portal client (Busch-Welcome apps): Used to inform the portal client a started surveillance call.

Usage	TO BE DISCUSSED:
	Is it required, that the portal server will also be informed, if a surveillance call was started from a local client (Comfort-Panel, indoor audio/video station, locally connec-ted mobile device)?
	If this is the case, we propose the following payload data for the event:
	{
	"client_local": "Friendly name of client - specified within the IP-Gateway?"
	}
	If the client who started the surveillance call is a remote client (ISPf portal client, e.g. remotely connected Busch-Welcome app), the IP-Gateway instead should inclu-de the UUID of the portal client:
	{
	"client_remote": "UUID of portal client"
	}

Status	Open

IS-880 - Ring – com.abb.ispf.event.welcome.ring

Direction	Bi-Directional
Purpose	Portal client (Busch-Welcome IP-Gateway) to portal server: Used to notify the portal server about a door ring. Portal server to portal client (Busch-Welcome apps): Used to inform the portal client about a door ring.
Usage	
Status	Open

IS-881 - Missed call – com.abb.ispf.event.welcome.call-missed

Direction	Bi-Directional
Purpose	Portal client (Busch-Welcome IP-Gateway) to portal server: Used to notify the portal server about a missed call.

	Portal server to portal client (Busch-Welcome apps): Used to inform the portal client about a missed call.
Usage	
Status	Open

IS-882 - Call answered – com.abb.ispf.event.welcome.call-answered

Direction	Bi-Directional
Purpose	Portal client (Busch-Welcome IP-Gateway) to portal server: Used to inform the portal server, that a door call has been answered. Portal server to portal client (Busch-Welcome apps): Used to inform the portal client about an answered call.

Usage	Portal client (Busch-Welcome IP-Gateway) to portal server:
	
	The portal client should specify in the payload, by which remote station (Comfort-Panel, indoor audio/video station, remote mobile device, local mobile device, ...) the call has been answered.
	
	If the call was answered by a ISPf portal client, the ID/UUID of the portal client should be embedded within the payload data.
	
	Proposal for payload data, if call was answered locally:
	{
	    "recipient_local": "Friendly name of client - specified within the IP-Gateway?"
	}
	
	Proposal for payload data, if call was answered remotely:
	{
	    "recipient_remote": "UUID of portal client"
	}
	
	The event must include a reference (belongsto) to the previously occured door ring event (com.abb.ispf.event.welcome.ring).

Status	Open

IS-883 - Hang up – com.abb.ispf.event.welcome.call-terminated

Direction	Bi-Directional
Purpose	Portal client (Busch-Welcome IP-Gateway) to portal server: Used to inform the portal server about a terminated call. Portal server to portal client (Busch-Welcome apps): Used to inform the portal client about a terminated call.

Usage	Portal client (Busch-Welcome IP-Gateway) to portal server: The event must include a reference (belongsto) to the previously occured event (com.abb.ispf.event.welcome.call-answered, com.abb.ispf.event.welcome.call-surveillance).
Status	Open

IS-884 - Screenshot – com.abb.ispf.event.welcome.screenshot

Direction	Bi-Directional
Purpose	Portal client (Busch-Welcome IP-Gateway) to portal server: Used to push a new screenshot to the portal server. Portal server to portal client (Busch-Welcome apps): Used to transmit a screenshot to a portal client.
Usage	Portal client (Busch-Welcome IP-Gateway) to portal server: The event must include a reference (belongsto) to a previously occured event: - ID of ring event (com.abb.ispf.event.welcome.ring) for automatically taken screenshots - ID of answered call event (com.abb.ispf.event.welcome.call-answered) for manually taken screenshots during call - ID of surveillance call event (com.abb.ispf.event.welcome.call-surveillance) for manually taken screenshots during call
Status	Open

IS-885 - Switch light – com.abb.ispf.event.welcome.light

Direction	Bi-Directional
Purpose	Portal client (Busch-Welcome IP-Gateway) to portal server: Used to inform the portal server that the light has been switched on or off. Portal server to portal client (Busch-Welcome apps): Used to inform the portal client that the light has been switched.

Usage	Portal client (Busch-Welcome IP-Gateway) to portal server:
	
	We propose the following payload data, if the light has been switched on:
	{
	"status": true
	}
	We propose the following payload data, if the light has been switched off:
	{
	"status": false
	}

Status	Open

IS-886 - Open door – com.abb.ispf.event.welcome.door-open

Direction	Client (IP-Gateway) to Server
Purpose	Used to inform the portal server, that the door has been opened.

Usage	The event should include from which client the door has been opened (Comfort-Panel, indoor audio/video station, locally connected mobile device, remotely connec-ted mobile device).
	Therefore, we propose the following payload data, if the client is locally connected to the IP-Gateway:
	{
	"client_local": "Friendly name of client - specified within the IP-Gateway?"
	}
	If the client is remotely connected, we propose the following payload data:
	{
	"client_remote": "UUID of portal client"
	}
	TO BE DISCUSSED:
	Is it possible to open the door via mobile devices without an active call? If not, we propose, that the event should include a reference (belongsto) to a previously event (com.abb.ispf.event.welcome.call-surveillance or com.abb.ispf.event.welcome.call-answered).

Status	Open