

# TruPortal™ 1.72 SDK 1.1.55 Guide

DOC-1105-EN-US Release Date: 4/5/2017 Revision: 05.02

## Contents

Overview .....	1
Contents .....	1
Release Notes .....	2
Technology .....	2
Documentation.....	2
List of Provided APIs .....	2
Examples .....	3
TruPortal REST API.....	4
TruPortal REST API Documentation .....	94
End User License Agreement for TruPortal SDK .....	107
Contact Information .....	109

## Overview

**NOTES:** SDK version 1.1.55 is compatible only with TruPortal version 1.72. SDK version 1.1.54 is compatible only with TruPortal version 1.71. Older panel versions (1.6 and 1.6SP1) require SDK version 1.1.51.

The TruPortal SDK was created to facilitate the integration between TruPortal and other systems. It is designed to allow third party applications or devices to perform the following types of operations:

- Retrieve events from the panel.
- Retrieve the current state of doors, readers, inputs, or outputs.
- Operate doors (lock down, unlock, momentary unlock, etc.).
- Execute action triggers.
- Change the state of an output.
- Manage cardholders (i.e. add, modify, or delete persons and badges; upload a picture for a person; change attributes of a badge, such as the assigned PIN, access levels, etc.), access levels, and schedules.
- Get a list of holidays, user-defined fields, action triggers, readers, reader groups, inputs, or outputs configured.
- Manage access permissions (i.e. add, modify, or delete access levels, assign/remove readers and reader groups in access levels).
- Manage schedules (i.e. add, modify, or delete time windows with weekday and holiday assignments).
- Log a custom event in the TruPortal event log.

For the list of individual APIs, refer to the **List of Provided APIs** table in this document.

## Contents

The TruPortal SDK consists of the following elements:

- A REST API plugin. TruPortal System Controllers must have this plugin installed for third party integration. The plugin can only be installed on TruPortal System Controllers with firmware version 1.6 or later.
- A set of 48 individual APIs, or more specifically, REST web services, that are exposed by the plugin.
- An interactive API reference. This web interface, available on a plugin-enabled System Controller via the URL, <https://mypanel.ip.address/api>, serves as API documentation and facilitates hands-on learning.
- A set of C# sample applications that illustrate how to use the individual REST web services in a C# .NET environment.

## Release Notes

### Version 1.1.55:

- Door command in POST /devices/doors/{id}/state request is extended to accept new values: grant-access, open, unlock, secure, lockout, reinstate
- Fix for capability of adding persons
- Fix for “empty” person photos created on adding persons

### Version 1.1.54:

- A “long polling” communication pattern is possible while getting Events. If there are no events to be reported, the API may delay the response up to 60 seconds, waiting for any data to be returned. Refer to the “GET /events” section for more details.
- Information about linked cameras is returned for relevant objects (devices, doors, readers, inputs, outputs).

### Version 1.1.53:

- The SDK package was updated to be compatible with panel firmware 1.71 and 1.72.  
**NOTE:** This version of the SDK is compatible only with TruPortal 1.71 and 1.72. Older panel versions (1.6 and 1.6SP1) require SDK version 1.1.51.

### Version 1.1.51:

- “/accessLevels” API extension – access levels now can be added, modified, or deleted.
- “/schedules” API introduction – schedules can be retrieved, added, modified, or deleted.
- “/holidays” API introduction – read-only option with minimal data (ID and name).
- “/devices/readers/groups” API introduction – read-only option with minimal data (ID and name).
- “/readers” API extension – two new parameters (doorId, direction) added.

## Technology

The technology used for this API is REST and JSON. This is a standard and widely-used web services technology. Therefore, to initiate a request, such as open a door, issue certain HTTP “GET” or “POST” requests.

For example, to momentarily unlock the door with an ID of “4”, one would issue a “HTTP POST” request to the following URL:

<https://mypanel.mycompany.com/api/devices/doors/4/state?command=grant-access>

To ensure security is not compromised, the API uses the same secure HTTP connection for accessing the web interface. Additionally, the client (third party software package) that is using the API must first log in using a username and password configured in the TruPortal system. The API client can only access functionality that has been granted to this user via the standard TruPortal operator roles. For example, if that user has only been granted read access to information, then opening a door via the API would not be allowed.

## Documentation

The majority of the documentation is provided in the form of an interactive reference which can directly use the APIs. The interactive reference also describes the parameters required for each API as well as the structure of the response.

In addition, there are several example applications written in C# .NET illustrating how to use the API from within another software package.

## List of Provided APIs

#	Category	API URL	Comment
1.1	Authentication	POST /auth/login	Log in
1.2		POST /auth/logout	Log out
2.1	Events	GET /events	Get all historical events
2.2		POST /events	Add (i.e. log) a new event
2.3		GET /events/{id}	Get a specific individual event by ID
3.1	Persons	GET /persons	Get all persons
3.2		POST /persons	Add a new person
3.3		GET /persons/{id}	Get a specific existing person by ID
3.4		PUT /persons/{id}	Modify an existing specific person
3.5		DELETE /persons/{id}	Delete a specific person

3.6		GET /persons/{id}/picture	Get the picture for a specific person
3.7		PUT /persons/{id}/picture	Upload a picture for a specific person
4.1	Credentials	GET /credentials	Get all credentials
4.2		POST /credentials	Add a new credential
4.3		GET /credentials/{id}	Get a specific existing credential by ID
4.4		PUT /credentials/{id}	Modify an existing credential
4.5		DELETE /credentials/{id}	Delete a specific credential
5.1	Access Level	GET /accessLevels	Get a list of all access levels
5.2		POST /accessLevels	Add a new access level
5.3		GET /accessLevels/{id}	Get one specific access level by ID
5.4		PUT /accessLevels/{id}	Modify an existing access level
5.5		DELETE /accessLevels/{id}	Delete a specific access level
6.1	Schedule	GET /schedules	Get a list of all schedules
6.2		POST /schedules	Add a new schedules
6.3		GET /schedules/{id}	Get one specific schedule by ID
6.4		PUT /schedules/{id}	Modify an existing schedule
6.5		DELETE /schedules/{id}	Delete a specific schedule
7.1	Holidays	GET /holidays	Get a list of all holidays
7.2		GET /holidays/{id}	Get one specific holiday by ID
8.1	User Defined Fields	GET /userDefinedFields	Get all user defined fields
8.2		GET /userDefinedFields/{id}	Get one specific user defined field
9.1	Action Triggers	GET /actionTriggers	Get a list of all action triggers
9.2		GET /actionTriggers/{id}	Get one specific action trigger
9.3		POST /actionTriggers/{id}/execute	Execute a specific action trigger
10.1	Devices	GET /devices	Get a list of all devices
10.2		GET /devices/{id}	Get one specific device
11.1	Doors	GET /devices/doors	Get a list of all doors
11.2		GET /devices/doors/{id}	Get one specific door
11.3		POST /devices/doors/{id}/state	Issue a door command
12.1	Readers	GET /devices/readers	Get a list of all readers
12.2		GET /devices/readers/{id}	Get one specific reader
12.3		GET /devices/readers/groups	Get a list of all reader groups
12.4		GET /devices/readers/groups/{id}	Get one specific reader group
13.1	Inputs	GET /devices/inputs	Get name and state of all inputs
13.2		GET /devices/inputs/{id}	Get one specific input
14.1	Outputs	GET /devices/outputs	Get name and state of all outputs
14.2		GET /devices/outputs/{id}	Get one specific output
14.3		POST /devices/outputs/{id}/state	Activate or deactivate an output

## Examples

The following examples further illustrate how individual APIs can be used to implement certain use cases. References to numbers, such as “#2.1,” refer to a specific API from the List of Provided APIs table in the previous section.

- Time and attendance reporting: a third party time and attendance software package can use the API to obtain the relevant access history information from a TruPortal System Controller
  - The key API for this would be GET /events (#2.1). All access history is available through historical events. Depending on how the third party package works, some of the other APIs might get used as well (for example, to obtain the complete list of cardholders).
- Initiating a complex action (such as lock-down, for example) from a third party software package
  - Suppose the customer’s management interface of choice is third party video monitoring software in which cameras are monitoring doors. If that video monitoring software has some form of macro, script, or other form of customization capabilities built in, it should be possible to add a button or menu item that is programmed for interaction with TruPortal. The button could be programmed to initiate a lock-down in TruPortal. To do that, one would set up an action trigger in TruPortal and then use the API, POST /actionTriggers/{id}/execute (#9.3) to

activate that action trigger. In response to this, not only may some or all doors get locked out, but TruPortal can be configured to send an e-mail. Previously, this would have required a hard-wired connection from the third party system into a TruPortal input. POST /events (#2.2) can be used to provide additional audit trail information indicating that the action trigger was activated by an API client rather than by a TruPortal application user.

- Integrating with a third party device that makes its own access decision
  - Suppose that a license plate recognition (LPR) system is unable to provide a Wiegand output and that hard-wiring of a relay output from that system into a TruPortal input is not practical. The system can use POST /devices/doors/{id}/state (#11.3) to open a door or gate when a license plate has been recognized. The API also allows attachment of a person name or card number to this command, so that this information becomes a permanent part of the event history. In addition, the LPR system might choose to use POST /events (#2.2) to log a custom event to provide additional audit trail information (see screenshot below).
- Data Integration between a human resources or IT system and TruPortal
  - Using POST /persons (#3.2), a third party software package can populate the TruPortal cardholder database with information from another database. This can be done in real time, i.e. as soon as a person is deleted from the human resources master database, the third party system can initiate DELETE /persons/{id} (#3.5) to delete that person from the TruPortal database (or perhaps just deactivate the person's badge by setting the "active to" date to the appropriate value).
  - A third party system can also use the APIs to learn about any changes to the TruPortal cardholder database, and then replicate the change.
- Initiating a momentary door unlock command from a third party device
  - A button on an intercom base station is used to open a specific door, i.e. the door at which somebody is speaking into a second intercom station. Typically, a hard-wired connection between the intercom client station at the door and TruPortal is established for the intercom to be able to open that specific door. Using the TruPortal SDK, this can now be done via the API, and because of POST /events (#2.2), additional audit trail information can be added to the event log that is not available in a hard-wired setup.

The following screenshot illustrates the use of POST /events (#2.2), as well as the ability to associate an "open door" command with a specific card/cardholder.

Event Description (Event Code)	Device	Person	Date/Time
Door Momentarily Unlocked (5824)	Door 8888-1	Netter, Christian (9347127)	10/2/2014 9:45:32 AM
License Plate Recognized (14591)	North gate	Netter, Christian	10/2/2014 9:45:21 AM

## TruPortal REST API

The TruPortal REST API documentation is included in this document. The example parameters are filled-in and the responses are shown for the following APIs only:

- POST /auth/login
- GET /events
- POST /events
- GET /persons
- GET /credentials



# auth : Operations for authentication

Show/Hide

List Operations

Expand Operations

Raw

POST

/auth/login

Log into the API

## Implementation Notes

This webservice call logs you into the TruPortal REST API.

You need to be logged into the API before you start calling API services. Sending valid credentials to this service will respond with an authenticated **Session Key** (formatted as e.g. **G6TAIjp6PqwWv4gOJkk0cPCeRShCDa60qe7ud5YI+K0=**).

This **Session Key** must be sent to all API service calls, passed within the **Authorization** header. To send the key when using the **Try it out!** buttons on this page, copy and paste the text into the **Session Key** text field at the top of this page.

The **Session Key** will timeout after **10 Minutes** of inactivity.

[List of response messages.](#)

## Response Class

Model

Model Schema

**SessionKey {**  
    **sessionKey** (*string*): This key represents the session for the authenticated user  
**}**

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>body</b>	<pre>{   "username": "admin",   "password": "demo" }</pre> <div>Parameter content type: application/json</div>	<b>Credentials to log into the system</b>	body	<div>ModelModel Schema</div> <b>LoginCredentials {</b> <b>username</b> ( <i>string</i> ), <b>password</b> ( <i>string</i> ) <b>}</b>

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/auth/login

## Response Body

```
{
  "sessionKey": "3y7Ac49V1xUN97eEvC2NXpp0pbxgfFbUq1CYH+shRhE="
}
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Thu, 21 Oct 2010 22:25:49 GMT",
  "Content-Type": "application/json",
  "Content-Length": "62",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]



## auth : Operations for authentication

[Show/Hide](#)[List Operations](#)[Expand Operations](#)[Raw](#)

POST

`/auth/logout`[Log out of the API](#)

### Implementation Notes

This webservice call logs you out from the TruPortal REST API.

Your current **Session Key** will become invalidated and no longer work with future service calls.

[Try it out!](#)[Hide Response](#)

### Request URL

`https://10.66.114.113:443/api/auth/logout`

### Response Body

`no content`

### Response Code

`200`

### Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:54:40 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "0",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: `https://10.66.114.113/api-docs` , API VERSION: 1.0.0 ]

## events : Operations about events

Show/Hide

List Operations

Expand Operations

Raw

GET

/events

Get all events

### Implementation Notes

Returns a list of **Events** on the TruPortal system, ordered chronologically from newest to oldest or from oldest to newest.

This service can only return up to **500** of the system events at one time. The **start** parameter can be used to specify the id of the event to begin with, if the parameter is not specified the most recent event will be used. If the **start** parameter is specified together with **reverse** parameter set to **true**, the events will be ordered from **start** to the newest. The **limit** parameter can be used to specify the number of events to be returned from the service call, only values between **1** and **500** are allowed.

If there are no events to be returned, the query waits for at most **timeout** seconds, and does internal polling for new events. If any new events appear before expiration of **timeout**, they are returned immediately. Otherwise, empty result is returned

The events returned may be optionally filtered by passing the **type**, **personId**, or **deviceId** parameters. Passing any of these will filter the results to only contain events of the specified type or id.

The resulting events will contain a localized string in the **description** field. The locale used for this field can be changed from the default of **en\_US** by setting the **locale** query parameter.

[List of response messages.](#)

### Response Class

Model

Model Schema

```
Event {
  id (integer): Unique identifier for the event,
  type (integer): Code representing the event type (see list of possible values),
  description (string): Localized description of the event type,
  personId (integer): Unique identifier for the person this event refers to. A value of 0 means no person,
  personName (string): Name of the person specified in the deviceId,
  deviceId (integer): Unique identifier for the device this event refers to. A value of 0 means no device,
  deviceName (string): Name of the device specified in the deviceId,
  timestamp (integer): The time this event occurred, represented by seconds since the Unix epoch
}
```

Response Content Type

application/json

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text" value="2"/>	Number of events to return (maximum 500)	query	integer
start	<input type="text"/>	Event id to start the results with, default is the most recent event	query	integer

Parameter	Value	Description	Parameter Type	Data Type
reverse	<input type="checkbox"/>	Reverse direction of events searching. If 'true', events with indexes greater or equal to 'start' will be returned, in ascending order. Otherwise events with indexes lower or equal to 'start' will be returned, in descending order.	query	boolean
locale	<input type="text"/>	Locale used to resolve the names (see <a href="#">list of possible values</a> )	query	string
type	<input type="text"/>	Event type to filter with (see <a href="#">list of possible values</a> )	query	integer
personId	<input type="text"/>	Filter to only events belonging to this person	query	integer
deviceId	<input type="text"/>	Filter to only events belonging to this device	query	integer
timeout	<input type="text"/>	Timeout in seconds for long polling. If there are no events to be returned, the API retries the query internally once per second, until any new events appear or timeout expires. The timeout may be specified in the range of 0-60 seconds (default value is 0, i.e. immediate response)	query	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/events?limit=2

## Response Body

```
[
  {
    "deviceName": "Controller (1)",
    "description": "Memory Backup Battery Low",
    "personName": "",
    "timestamp": 1287690175,
    "personId": 0,
    "type": 14618,
```

```
"id": 159,  
  "deviceId": 1  
},  
{  
  "deviceName": "Controller (1)",  
  "description": "Memory Backup Battery Low",  
  "personName": "",  
  "timestamp": 1287668575,  
  "personId": 0,  
  "type": 14618,  
  "id": 158,  
  "deviceId": 1  
}  
]
```

## Response Code

200

## Response Headers

```
{  
  "Server": "",  
  "Date": "Thu, 21 Oct 2010 22:33:48 GMT",  
  "Content-Type": "application/json",  
  "Content-Length": "354",  
  "Connection": "keep-alive",  
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",  
  "X-Content-Type-Options": "nosniff",  
  "Cache-Control": "no-cache, no-store, private",  
  "X-Xss-Protection": "1; mode=block"  
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]



# events : Operations about events

Show/Hide

List Operations

Expand Operations

Raw

POST

/events

Add a new event

## Implementation Notes

Adds a new **Event** to the system. The event **type** code of the new event will be set to **14591**.

The **description** field of the new event must be specified, it determines the string that will be displayed in the **Event Description** column from the TruPortal events view. The **deviceDescription** and **personDescription** fields can optionally be set to display additional information in the **Device** and **Person** column of the TruPortal events view.

This service returns the unique identifier for the newly added event.

[List of response messages.](#)

## Response Class

Model

Model Schema

integer

Response Content Type 

text/plain

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	<div><pre>{   "description": "test",   "deviceDescription": "3rd party device",   "personDescription": "John Doe" }</pre></div> <div>Parameter content type: <div>application/json</div></div>	<b>Event to add to the system</b>	body	<div>Model</div> <div>Model Schema</div> <div><b>Event {</b> <b>description</b> (<a href="#">string</a>), <b>deviceDescription</b> (<a href="#">string</a>, optional), <b>personDescription</b> (<a href="#">string</a>, optional) <b>}</b></div>

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/events

## Response Body

161

## Response Code

200

## Response Headers

{

```
"Server": "",  
"Date": "Thu, 21 Oct 2010 22:41:42 GMT",  
"Content-Type": "text/html; charset=UTF-8",  
"Content-Length": "3",  
"Connection": "keep-alive",  
"Strict-Transport-Security": "max-age=31536000; includeSubDomains",  
"X-Content-Type-Options": "nosniff",  
"Cache-Control": "no-cache, no-store, private",  
"X-Xss-Protection": "1; mode=block"  
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

## events : Operations about events

Show/Hide

List Operations

Expand Operations

Raw

GET	/events/{id}	Get an event by id
-----	--------------	--------------------

### Implementation Notes

Returns a single **Event**, specified by the id in the url.

The resulting event will contain a localized string in the **description** field. The locale used for this field can be changed from the default of **en\_US** by setting the **locale** query parameter.

If there is no event to be returned, the query waits for at most **timeout** seconds, and does internal polling for new events. If a new event with specified id appears before expiration of **timeout**, it is returned immediately. Otherwise, empty result is returned

[List of response messages.](#)

### Response Class

Model

Model Schema

**Event {**

- id** (*integer*): Unique identifier for the event,
- type** (*integer*): Code representing the event type (see [list of possible values](#)),
- description** (*string*): Localized description of the event **type**,
- personId** (*integer*): Unique identifier for the person this event refers to. A value of **0** means no person,
- personName** (*string*): Name of the person specified in the **deviceId**,
- deviceId** (*integer*): Unique identifier for the device this event refers to. A value of **0** means no device,
- deviceName** (*string*): Name of the device specified in the **deviceId**,
- timestamp** (*integer*): The time this event occurred, represented by seconds since the **Unix epoch**

**}**

Response Content Type

application/json

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="161"/>	<b>Unique identifier of the event to return</b>	path	integer
locale	<input type="text"/>	Locale used to resolve the names (see <a href="#">list of possible values</a> )	query	string
timeout	<input type="text"/>	Timeout in seconds for long polling. If there is no event to be returned, the API retries the query internally once per second, until a new event appears or timeout expires. The timeout may be specified in the range	query	integer

Parameter	Value	Description	Parameter Type	Data Type
		of 0-60 seconds (default value is 0, i.e. immediate response)		

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/events/161

### Response Body

```
{
  "deviceName": "3rd party device",
  "description": "test",
  "personName": "John Doe",
  "timestamp": 1287700902,
  "personId": 0,
  "type": 14591,
  "id": 161,
  "deviceId": 0
}
```

### Response Code

200

### Response Headers

```
{
  "Server": "",
  "Date": "Thu, 21 Oct 2010 22:54:51 GMT",
  "Content-Type": "application/json",
  "Content-Length": "164",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

# persons : Operations about persons

Show/Hide

List Operations

Expand Operations

Raw

GET

/persons

Get all persons

## Implementation Notes

Returns a list of **Persons** in the TruPortal system.

This service can only return up to **250** of the persons at one time. The **start** parameter can be used to specify the id of the person to begin with, if the parameter is not specified the first person will be used. The **limit** parameter can be used to specify the number of persons to be returned from the service call, only values between **1** and **250** are allowed.

The persons returned can optionally be filtered by passing the **firstname**, **middlename**, or **lastname** parameters. These search parameters will filter the results to include only persons whose names begin with the values provided.

The person objects returned may contain a list of **userDefinedFields**. This list will contain an entry for each field that has data, if the **data** field is empty it will be excluded from the response. If a field is **secured** and the authenticated user does not have permissions to view secured fields it will also be excluded from the response.

[List of response messages.](#)

## Response Class

Model

Model Schema

```
Person {
  id (integer, optional): unique identifier for the person. (ignored when adding or modifying),
  firstname (string): first name of the person,
  middlename (string): middle name of the person,
  lastname (string): last name of the person,
  userDefinedFields (array[CustomPersonData], optional): collection of the data entered into the user defined fields
}

CustomPersonData {
  id (integer): unique identifier to determine which field this data belongs to,
  data (string): the data for this user defined field
}
```

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text"/>	Number of persons to return (maximum 250)	query	integer
start	<input type="text"/>	Person id to start the results with, default is the first person on the system	query	integer
firstName	<input type="text" value="John"/>	Search for persons that start with this first name	query	string

Parameter	Value	Description	Parameter Type	Data Type
middleName	<input type="text"/>	Search for persons that start with this middle name	query	string
lastName	<input type="text"/>	Search for persons that start with this last name	query	string

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/persons?firstName=John

### Response Body

```
[
  {
    "middlename": "f",
    "lastname": "Doe",
    "userDefinedFields": [],
    "id": 14,
    "firstname": "John"
  }
]
```

### Response Code

200

### Response Headers

```
{
  "Server": "",
  "Date": "Thu, 21 Oct 2010 23:05:31 GMT",
  "Content-Type": "application/json",
  "Content-Length": "96",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```



# persons : Operations about persons

Show/Hide

List Operations

Expand Operations

Raw

POST

/persons

Add a new person

## Implementation Notes

Adds a new person to the system.

At least one of the name fields (e.g. **firstname**, **middlename**, or **lastname**) must be set for the data structure to be valid. The **userDefinedFields** parameter is optional, if it is passed the ids within it must not be duplicated.

This service will generate a unique identifier for the newly added person and return it as the response. Any **id** passed within the body structure will be ignored.

[List of response messages.](#)

## Response Class

Model

Model Schema

integer

Response Content Type 

text/plain

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	<pre>{   "id": 0,   "firstname": "John",   "middlename": "f",   "lastname": "Doe",   "userDefinedFields": [     {       "id": 0,       "data": " "     }   ] }</pre>	Person object to add	body	<div>Model</div> <div>Model Schema</div> <div><b>Person {</b>   <b>id</b> (<b>integer</b>, optional): unique identifier for the person. (ignored when adding or modifying),   <b>firstname</b> (<b>string</b>): first name of the person,   <b>middlename</b> (<b>string</b>): middle name of the person,   <b>lastname</b> (<b>string</b>): last name of the person,   <b>userDefinedFields</b> (<b>array[CustomPersonData]</b>, optional): collection of the <b>data</b> entered into the user defined fields <b>}</b> <b>CustomPersonData {</b>   <b>id</b> (<b>integer</b>): unique identifier to determine which field this data belongs to,   <b>data</b> (<b>string</b>): the data for this user defined field <b>}</b></div>

Parameter content type: 

application/json

## Request URL

https://10.66.114.113:443/api/persons

## Response Body

14

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Thu, 21 Oct 2010 23:03:10 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "2",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: https://10.66.114.113/api-docs , API VERSION: 1.0.0 ]

# persons : Operations about persons

Show/Hide

List Operations

Expand Operations

Raw

GET

/persons/{id}

Get a person by id

## Implementation Notes

Returns a single **Person**, specified by the **id** in the url.

The person object returned may contain a list of **userDefinedFields**. This list will contain an entry for each field that has data, if the **data** field is empty it will be excluded from the response. If a field is **secured** and the authenticated user does not have permissions to view secured fields it will also be excluded from the response.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Person {**

- id** (*integer*, optional): unique identifier for the person. (ignored when adding or modifying),
- firstname** (*string*): first name of the person,
- middlename** (*string*): middle name of the person,
- lastname** (*string*): last name of the person,
- userDefinedFields** (*array[CustomPersonData]*, optional): collection of the **data** entered into the user defined fields

**}**

**CustomPersonData {**

- id** (*integer*): unique identifier to determine which field this data belongs to,
- data** (*string*): the data for this user defined field

**}**

Response Content Type 

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<div>14</div>	Unique identifier of the person to return	path	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/persons/14

## Response Body

```
{
  "middlename": "f",
  "lastname": "Doe",
  "userDefinedFields": [],
  "id": 14,
  "firstname": "John"
```

```
}
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Thu, 21 Oct 2010 23:08:29 GMT",
  "Content-Type": "application/json",
  "Content-Length": "94",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

persons : Operations about persons

Show/Hide | List Operations | Expand Operations | Raw

PUT

/persons/{id}

Modify an existing person

Implementation Notes

Modifies an existing person on the system, specified by the **id** in the url.

At least one of the name fields (e.g. **firstname**, **middlename**, or **lastname**) must be set for the data structure to be valid. The **userDefinedFields** parameter is optional, if it is passed the ids within it must not be duplicated.

[List of response messages.](#)

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	14	Unique identifier of the person to modify	path	integer
body	<pre>{   "id": 0,   "firstname": "Jonathan",   "middlename": "F",   "lastname": "Doe",   "userDefinedFields": [     {       "id": 0,       "data": ""     }   ] }</pre>	Person object to modify	body	Model   Model Schema <b>Person {</b> <b>id</b> ( <a href="#">integer</a> , optional): unique identifier for the person. (ignored when adding or modifying), <b>firstname</b> ( <a href="#">string</a> ): first name of the person, <b>middlename</b> ( <a href="#">string</a> ): middle name of the person, <b>lastname</b> ( <a href="#">string</a> ): last name of the person, <b>userDefinedFields</b> ( <a href="#">array[CustomPersonData]</a> , optional): collection of the <b>data</b> entered into the user defined fields <b>}</b> <b>CustomPersonData {</b> <b>id</b> ( <a href="#">integer</a> ): unique identifier to determine which field this data belongs to, <b>data</b> ( <a href="#">string</a> ): the data for this user defined field <b>}</b>

Parameter content type: application/json

Try it out!

Hide Response

Request URL

https://10.66.114.113:443/api/persons/14

Response Body

no content

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:00:06 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "0",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]



persons : Operations about persons

Show/Hide | List Operations | Expand Operations | Raw

DELETE

/persons/{id}

Delete a person by id

### Implementation Notes

Permanently removes a person from the system, specified by the **id** in the url.

[List of response messages.](#)

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="14"/>	Unique identifier of the person to delete	path	integer

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/persons/14

### Response Body

no content

### Response Code

204

### Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:03:28 GMT",
  "Content-Length": "55",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

# persons : Operations about persons

Show/Hide

List Operations

Expand Operations

Raw

GET

/persons/{id}/picture

Get the picture for a person

## Implementation Notes

Gets the picture for a person on the system, specified by the **id** in the url.

The picture will be returned as binary JPEG data.

[List of response messages.](#)

## Response Class

Model

Model Schema

File

Response Content Type

image/jpeg

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1	Unique identifier of the person whose picture will be returned	path	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/persons/1/picture

## Response Body

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:06:07 GMT",
  "Content-Type": "image/jpeg",
  "Content-Length": "2641",
  "Connection": "keep-alive",
  "Last-Modified": "Thu, 21 Oct 2010 05:39:45 GMT",
  "Accept-Ranges": "bytes",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

persons : Operations about persons

Show/Hide | List Operations | Expand Operations | Raw

PUT

/persons/{id}/picture

Uploads a picture for a person

### Implementation Notes

Sets the picture for a person on the system, specified by the **id** in the url.

[List of response messages.](#)

### Response Class

string

Response Content Type 

text/plain

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<div>1</div>	Unique identifier of the person whose picture will be set	path	integer
picture	<div><div>Przeglądaj...</div>Untitled.jpg</div>	Picture to upload	form	File

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/persons/1/picture

### Response Body

no content

### Response Code

200

### Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "0",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "x-xss-protection": "1; mode=block"
}
```

# credentials : Operations about credentials

Show/Hide

List Operations

Expand Operations

Raw

GET

/credentials

Get all credentials

## Implementation Notes

Returns a list of **Credentials** in the TruPortal system.

This service can only return up to **250** of the system credentials at one time. The **start** parameter can be used to specify the id of the credential to begin with. If the parameter is not specified, the first credential will be used. The **limit** parameter can be used to specify the number of credentials to be returned from the service call; only values between **1** and **250** are allowed.

[List of response messages.](#)

## Response Class

Model

Model Schema

### Credential {

**id** ([integer](#), optional): Unique identifier for the credential used in the TruPortal system,

**personId** ([integer](#)): Identifier of person to whom the credential belongs,

**cardNumber** ([string](#)): Unique number of the credential set by its manufacturer,

**issueCode** ([integer](#), optional): Indicates the selected credential's issue code,

**pin** ([integer](#), optional): Personal identification number,

**activeFrom** ([string](#), optional): Time the credential is valid from (i.e. 2000-01-29 22:41:59),

**activeTo** ([string](#), optional): Time the credential is valid to (i.e. 2000-01-29 22:41:59),

**antipassbackExempt** ([boolean](#), optional): Anti-passback violation for the selected credential will grant access into the anti-passback area without a violation noted in Alarm Monitoring,

**extendedAccess** ([boolean](#), optional): Extended held open and extended strike times will be used for the selected credential,

**accessLevels** ([array\[integer\]](#), optional): List of access permissions for the credential

}

Response Content Type 

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text"/>	Number of credentials to return (maximum 250)	query	integer
start	<input type="text"/>	Credential id to start the result with	query	integer
personId	<input type="text" value="3"/>	Filter for credentials belonging to this person	query	integer
cardNumber	<input type="text"/>	Search for credentials with this card number	query	string

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/credentials?personId=3

## Response Body

```
[
  {
    "activeFrom": null,
    "antipassbackExempt": true,
    "cardNumber": "3",
    "accessLevels": [],
    "pin": 3,
    "personId": 3,
    "extendedAccess": false,
    "activeTo": "2020-07-01 23:59:59",
    "id": 3,
    "issueCode": 3
  },
  {
    "activeFrom": "2011-02-03 00:00:00",
    "antipassbackExempt": true,
    "cardNumber": "4",
    "accessLevels": [],
    "pin": 4,
    "personId": 3
  }
]
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:12:57 GMT",
  "Content-Type": "application/json",
  "Content-Length": "414",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

# credentials : Operations about credentials

Show/Hide

List Operations

Expand Operations

Raw

POST

/credentials

Add a new credential

## Implementation Notes

Add a new credential to the system and return id.

The **personId** and the **cardNumber** parameters both have to be fulfilled. The **personId** parameter identifies who is owner of the credential and **cardNumber** defines unique number of the credential defined by its manufacturer. Other parameters are optional. The **id** parameter is ignored, but id of newly created credential is returned.

[List of response messages.](#)

## Response Class

Model

Model Schema

integer

Response Content Type 

text/plain

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	<pre>{   "id": 0,   "personId": 2,   "cardNumber": "1234567",   "issueCode": 0,   "pin": 0,   "activeFrom": null,   "activeTo": null,   "antipassbackExempt": false,   "extendedAccess": false,   "accessLevels": [     0   ] }</pre>	<b>Credential object to add</b>	body	<div>ModelModel Schema</div> <div><b>Credential {</b> <b>id</b> (<i>integer</i>, optional): Unique identifier for the credential used in the TruPortal system, <b>personId</b> (<i>integer</i>): Identifier of person to whom the credential belongs, <b>cardNumber</b> (<i>string</i>): Unique number of the credential set by its manufacturer, <b>issueCode</b> (<i>integer</i>, optional): Indicates the selected credential's issue code, <b>pin</b> (<i>integer</i>, optional): Personal identification number, <b>activeFrom</b> (<i>string</i>, optional): Time the credential is valid from (i.e. 2000-01-29 22:41:59), <b>activeTo</b> (<i>string</i>, optional): Time the credential is valid to (i.e. 2000-01-29 22:41:59), <b>antipassbackExempt</b></div>



Parameter	Value	Description	Parameter Type	Data Type
				( <a href="#">boolean</a> , optional): Anti-passback violation for the selected credential will grant access into the anti-passback area without a violation noted in Alarm Monitoring, <b>extendedAccess</b> ( <a href="#">boolean</a> , optional): Extended held open and extended strike times will be used for the selected credential, <b>accessLevels</b> ( <a href="#">array[integer]</a> , optional): List of access permissions for the credential }

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/credentials

### Response Body

19

### Response Code

200

### Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:16:16 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "2",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

# credentials : Operations about credentials

Show/Hide

List Operations

Expand Operations

Raw

GET

/credentials/{id}

Get a credential by id

## Implementation Notes

Returns a single **credential**, specified by the id in the url.

The **id** has to be defined and valid identifier for requested credentials.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Credential** {

**id** ([integer](#), optional): Unique identifier for the credential used in the TruPortal system,

**personId** ([integer](#)): Identifier of person to whom the credential belongs,

**cardNumber** ([string](#)): Unique number of the credential set by its manufacturer,

**issueCode** ([integer](#), optional): Indicates the selected credential's issue code,

**pin** ([integer](#), optional): Personal identification number,

**activeFrom** ([string](#), optional): Time the credential is valid from (i.e. 2000-01-29 22:41:59),

**activeTo** ([string](#), optional): Time the credential is valid to (i.e. 2000-01-29 22:41:59),

**antipassbackExempt** ([boolean](#), optional): Anti-passback violation for the selected credential will grant access into the anti-passback area without a violation noted in Alarm Monitoring,

**extendedAccess** ([boolean](#), optional): Extended held open and extended strike times will be used for the selected credential,

**accessLevels** ([array\[integer\]](#), optional): List of access permissions for the credential

}

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	19	Unique identifier of the credential to return	path	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/credentials/19

## Response Body

```
{
  "activeFrom": null,
  "antipassbackExempt": false,
  "cardNumber": "1234567",
  "accessLevels": [],
  "pin": 0,
```

```
"personId": 2,  
"extendedAccess": false,  
"activeTo": null,  
"id": 19,  
"issueCode": 0  
}
```

## Response Code

200

## Response Headers

```
{  
  "Server": "",  
  "Date": "Fri, 22 Oct 2010 00:18:30 GMT",  
  "Content-Type": "application/json",  
  "Content-Length": "188",  
  "Connection": "keep-alive",  
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",  
  "X-Content-Type-Options": "nosniff",  
  "Cache-Control": "no-cache, no-store, private",  
  "X-Xss-Protection": "1; mode=block"  
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# credentials : Operations about credentials

Show/Hide

List Operations

Expand Operations

Raw

PUT

/credentials/{id}

Modify an existing credential

## Implementation Notes

Modifies an existing credential in the system, specified by the **id** in the url.

All properties will be overridden, even if the parameter (ie. activeFrom) is not defined, it will be changed to the default value. The exception to that rule is the **cardNumber** parameter, for which modification is not allowed.

[List of response messages.](#)

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	19	Unique identifier of the credential	path	integer
body	<pre>{   "id": 0,   "personId": 2,   "cardNumber": "55554444",   "issueCode": 0,   "pin": 0,   "activeFrom": null,   "activeTo": null,   "antipassbackExempt": false,   "extendedAccess": false,   "accessLevels": [     0   ] }</pre>	Credential object to modify	body	<div>ModelModel Schema</div> <pre>{   "id": 0,   "personId": 0,   "cardNumber": "",   "issueCode": 0,   "pin": 0,   "activeFrom": "",   "activeTo": "",   "antipassbackExempt": false,   "extendedAccess": false,   "accessLevels": [     0   ] }</pre> <div>Click to set as parameter value</div>

Parameter content type: application/json

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/credentials/19

## Response Body

no content

## Response Code

200

## Response Headers

{

```
"Server": "",  
"Date": "Fri, 22 Oct 2010 00:23:16 GMT",  
"Content-Type": "text/html; charset=UTF-8",  
"Content-Length": "0",  
"Connection": "keep-alive",  
"Strict-Transport-Security": "max-age=31536000; includeSubDomains",  
"X-Content-Type-Options": "nosniff",  
"Cache-Control": "no-cache, no-store, private",  
"X-Xss-Protection": "1; mode=block"  
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# credentials : Operations about credentials

Show/Hide

List Operations

Expand Operations

Raw

DELETE

/credentials/{id}

Delete a credential by id

## Implementation Notes

Permanently remove credential with **id** from the truPortal system.

[List of response messages.](#)

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="19"/>	Unique identifier of the credential to delete	path	integer

Try it out!

Hide Response

## Request URL

https://10.66.114.113:443/api/credentials/19

## Response Body

no content

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:25:42 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "0",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

accessLevels : Operations about access levels

Show/Hide | List Operations | Expand Operations | Raw

GET

/accessLevels

Get all access levels

### Implementation Notes

Returns a list of **AccessLevel** from the TruPortal system, ordered by id.

This service can only return up to **250** of the access levels at one time. The **start** parameter can be used to specify the id of the access level to begin with, if the parameter is not specified the first id (the smallest value) will be used. The **limit** parameter can be used to specify the number of access levels to be returned from the service call, only values between **1** and **250** are allowed.

[List of response messages.](#)

### Response Class

Model

Model Schema

```
AccessLevel {
  id (integer): Unique identifier for the access level,
  name (string): Name of the access level,
  readerGroupSchedules (array[ReaderGroupSchedule], optional): Collection of the ReaderGroupSchedule,
  readerSchedules (array[ReaderSchedule], optional): Collection of the ReaderSchedule
}

ReaderGroupSchedule {
  readerGroupId (integer): Unique identifier of the reader group,
  scheduleId (integer): Unique identifier for the schedule
}

ReaderSchedule {
  readerId (integer): Unique identifier of the reader,
  scheduleId (integer): Unique identifier for the schedule
}
```

Response Content Type

application/json

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text"/>	Number of access levels to return (maximum 250)	query	integer
start	<input type="text"/>	Access level id to start the results with	query	integer

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/accessLevels

### Response Body

```
[
  {
    "readerGroupSchedules": [],
    "id": 1,
    "readerSchedules": [
      {
        "readerId": 6,
        "scheduleId": 1
      }
    ],
    "name": "ac1"
  },
  {
    "readerGroupSchedules": [],
    "id": 2,
    "readerSchedules": [
      {
        "readerId": 7,
        "scheduleId": 1
      }
    ]
  }
]
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:29:43 GMT",
  "Content-Type": "application/json",
  "Content-Length": "361",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]



# accessLevels : Operations about access levels

Show/Hide | List Operations | Expand Operations | Raw

POST

/accessLevels

Add a new access level

## Implementation Notes

Adds a new access level to the system.

This service will generate a unique identifier for the newly added access level and return it as the response. Any **id** passed within the body structure will be ignored.

[List of response messages.](#)

## Response Class

Model

Model Schema

integer

Response Content Type

text/plain

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	<pre>{  "id": 4,  "name": "ac5"}</pre>	Access level object to add	body	<div>Model   Model Schema</div> <div><b>AccessLevel {</b>   <b>id</b> (<b>integer</b>): Unique identifier for the access level,   <b>name</b> (<b>string</b>): Name of the access level,   <b>readerGroupSchedules</b> (<b>array[ReaderGroupSchedule]</b>, optional): Collection of the <b>ReaderGroupSchedule</b>,   <b>readerSchedules</b> (<b>array[ReaderSchedule]</b>, optional): Collection of the <b>ReaderSchedule</b> <b>}</b> <b>ReaderGroupSchedule {</b>   <b>readerGroupId</b> (<b>integer</b>): Unique identifier of the reader group,   <b>scheduleId</b> (<b>integer</b>): Unique identifier for the schedule <b>}</b> <b>ReaderSchedule {</b>   <b>readerId</b> (<b>integer</b>): Unique identifier of the reader,   <b>scheduleId</b> (<b>integer</b>): Unique identifier for the schedule <b>}</b></div>

Parameter content type:

application/json

## Request URL

https://10.66.114.113:443/api/accessLevels

## Response Body

5

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:35:25 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "1",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: https://10.66.114.113/api-docs , API VERSION: 1.0.0 ]

# accessLevels : Operations about access levels

Show/Hide

List Operations

Expand Operations

Raw

GET

/accessLevels/{id}

Get an access level by id

## Implementation Notes

Returns a single **AccessLevel**, specified by the id in the url.

[List of response messages.](#)

## Response Class

Model

Model Schema

```
AccessLevel {
  id (integer): Unique identifier for the access level,
  name (string): Name of the access level,
  readerGroupSchedules (array[ReaderGroupSchedule], optional): Collection of the ReaderGroupSchedule,
  readerSchedules (array[ReaderSchedule], optional): Collection of the ReaderSchedule
}

ReaderGroupSchedule {
  readerGroupId (integer): Unique identifier of the reader group,
  scheduleId (integer): Unique identifier for the schedule
}

ReaderSchedule {
  readerId (integer): Unique identifier of the reader,
  scheduleId (integer): Unique identifier for the schedule
}
```

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="5"/>	Unique identifier of the access level to return	path	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/accessLevels/5

## Response Body

```
{
  "readerGroupSchedules": [],
  "id": 5,
  "readerSchedules": [],
  "name": "ac5"
}
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:37:47 GMT",
  "Content-Type": "application/json",
  "Content-Length": "75",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

accessLevels : Operations about access levels

Show/Hide | List Operations | Expand Operations | Raw

PUT

/accessLevels/{id}

Modify an existing access level

Implementation Notes

Modifies an existing access level on the system, specified by the **id** in the url.

[List of response messages.](#)

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	5	Unique identifier of the access level to modify	path	integer
body	<pre>{  "id": 5,  "name": "accesslevel5"}</pre> <div>Parameter content type: application/json</div>	Access level object to modify	body	<div>Model   Model Schema</div> <div><b>AccessLevel {</b>   <b>id</b> (<i>integer</i>): Unique identifier for the access level,   <b>name</b> (<i>string</i>): Name of the access level,   <b>readerGroupSchedules</b> (<i>array[ReaderGroupSchedule]</i>, optional): Collection of the <b>ReaderGroupSchedule</b>,   <b>readerSchedules</b> (<i>array[ReaderSchedule]</i>, optional): Collection of the <b>ReaderSchedule</b> <b>}</b> <b>ReaderGroupSchedule {</b>   <b>readerGroupId</b> (<i>integer</i>): Unique identifier of the reader group,   <b>scheduleId</b> (<i>integer</i>): Unique identifier for the schedule <b>}</b> <b>ReaderSchedule {</b>   <b>readerId</b> (<i>integer</i>): Unique identifier of the reader,   <b>scheduleId</b> (<i>integer</i>): Unique identifier for the schedule <b>}</b></div>

Try it out!

[Hide Response](#)

Request URL

https://10.66.114.113:443/api/accessLevels/5

Response Body

no content

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:41:57 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "0",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

## accessLevels : Operations about access levels

Show/Hide

List Operations

Expand Operations

Raw

DELETE

/accessLevels/{id}

Delete an access level by id

### Implementation Notes

Permanently removes an access level from the system, specified by the **id** in the url.

[List of response messages.](#)

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="5"/>	Unique identifier of the access level to delete	path	integer

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/accessLevels/5

### Response Body

no content

### Response Code

200

### Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:43:54 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "0",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

schedules : Operations about schedules

Show/Hide | List Operations | Expand Operations | Raw

GET

/schedules

Get all schedules

Implementation Notes

Returns a list of **Schedule** from the TruPortal system, ordered by id.

This service can only return up to **250** of the schedules at one time. The **start** parameter can be used to specify the id of the schedule to begin with, if the parameter is not specified the first id (the smallest value) will be used. The **limit** parameter can be used to specify the number of schedules to be returned from the service call, only values between **1** and **250** are allowed.

The resulting schedules may contain a localized string in the **name** field (only factory defined, not overridden fields are localized). The locale used for this field can be changed from the default of **en\_US** by setting the **locale** query parameter.

[List of response messages.](#)

Response Class

Model | Model Schema

**Schedule {**

- id** ([integer](#)): Unique identifier for the schedule,
- name** ([string](#)): Name of the schedule. May be localized or not (see implementation notes above),
- intervals** ([array\[TimeWindow\]](#)): Collection of the **TimeWindow**,
- holidays** ([array\[integer\]](#), optional): List of holidays assigned to the schedule

**}**

**TimeWindow {**

- daysOfTheWeek** ([array\[integer\]](#), optional): The days of the week on which to grant access,
- start** ([string](#)): The time during which the schedule will be active,
- end** ([string](#)): The time during which the schedule will be active

**}**

Response Content Type 

application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<div>2</div>	Number of schedules to return (maximum 250)	query	integer
start	<div></div>	Schedule id to start the results with	query	integer
locale	<div></div>	Locale used to resolve the names (see <a href="#">list of possible values</a> )	query	string

Try it out!

[Hide Response](#)

Request URL



## Response Body

```
[
  {
    "intervals": [
      {
        "start": "00:00",
        "daysOfTheWeek": [
          1,
          2,
          3,
          4,
          5,
          6,
          7
        ],
        "end": "23:59"
      }
    ],
    "id": 1,
    "holidays": [
      1,
      3,
      5,
      7
    ],
    "name": "Everyday 24/7"
  },
  {
    "intervals": [
      {
        "start": "08:00",
        "daysOfTheWeek": [
          2,
          3,
          4,
          5,
          6
        ],
        "end": "17:00"
      }
    ],
    "id": 2,
    "holidays": [
      3,
      4,
      5,
      6
    ],
    "name": "Weekdays 8AM-5PM"
  }
]
```

```
]

```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:48:52 GMT",
  "Content-Type": "application/json",
  "Content-Length": "308",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# schedules : Operations about schedules

Show/Hide

List Operations

Expand Operations

Raw

POST

/schedules

Add a new schedule

## Implementation Notes

Adds a new schedule to the system.

This service will generate a unique identifier for the newly added schedule and return it as the response. Any **id** passed within the body structure will be ignored.

[List of response messages.](#)

## Response Class

Model

Model Schema

integer

Response Content Type 

text/plain

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
body	<pre>{   "id": 0,   "name": "new schedule",   "intervals": [     {       "daysOfTheWeek": [         0       ],       "start": "12:00",       "end": "14:00"     }   ],   "holidays": [     1   ] }</pre>	Schedule object to add	body	<div>ModelModel Schema</div> <div><b>Schedule {</b> <b>id</b> (<a href="#">integer</a>): Unique identifier for the schedule, <b>name</b> (<a href="#">string</a>): Name of the schedule. May be localized or not (see implementation notes above), <b>intervals</b> (<a href="#">array[TimeWindow]</a>): Collection of the <b>TimeWindow</b>, <b>holidays</b> (<a href="#">array[integer]</a>, optional): List of holidays assigned to the schedule <b>}</b> <b>TimeWindow {</b> <b>daysOfTheWeek</b> (<a href="#">array[integer]</a>, optional): The days of the week on which to grant access, <b>start</b> (<a href="#">string</a>): The time during which the schedule will be active, <b>end</b> (<a href="#">string</a>): The time during which the schedule will be active <b>}</b></div>

Parameter content type: 

application/json

## Request URL

https://10.66.114.113:443/api/schedules

## Response Body

8

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:51:35 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "1",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: https://10.66.114.113/api-docs , API VERSION: 1.0.0 ]

# schedules : Operations about schedules

Show/Hide

List Operations

Expand Operations

Raw

GET

/schedules/{id}

Get a schedule by id

## Implementation Notes

Returns a single **Schedule**, specified by the id in the url.

The resulting schedule may contain a localized string in the **name** field (only factory defined, not overridden fields are localized). The locale used for this field can be changed from the default of **en\_US** by setting the **locale** query parameter.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Schedule {**

- id** ([integer](#)): Unique identifier for the schedule,
- name** ([string](#)): Name of the schedule. May be localized or not (see implementation notes above),
- intervals** ([array\[TimeWindow\]](#)): Collection of the **TimeWindow**,
- holidays** ([array\[integer\]](#), optional): List of holidays assigned to the schedule

**}**

**TimeWindow {**

- daysOfTheWeek** ([array\[integer\]](#), optional): The days of the week on which to grant access,
- start** ([string](#)): The time during which the schedule will be active,
- end** ([string](#)): The time during which the schedule will be active

**}**

Response Content Type 

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<div>8</div>	Unique identifier of the schedule to return	path	integer
locale	<div></div>	Locale used to resolve the names (see <a href="#">list of possible values</a> )	query	string

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/schedules/8

## Response Body

```
{
  "intervals": [
```

```
{
  "start": "12:00",
  "daysOfTheWeek": [],
  "end": "14:00"
},
{id": 8,
"holidays": [
  1
],
"name": "new schedule"
}
```

### Response Code

200

### Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:53:40 GMT",
  "Content-Type": "application/json",
  "Content-Length": "122",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: https://10.66.114.113/api-docs , API VERSION: 1.0.0 ]

schedules : Operations about schedules

Show/Hide | List Operations | Expand Operations | Raw

PUT

/schedules/{id}

Modify an existing schedule

Implementation Notes

Modifies an existing schedule on the system, specified by the **id** in the url.

[List of response messages.](#)

Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="8"/>	Unique identifier of the schedule to modify	path	integer
body	<div><pre>{   "id": 0,   "name": "schedule new name",   "intervals": [     {       "daysOfTheWeek": [         0       ],       "start": "16:00",       "end": "17:00"     }   ],   "holidays": [     1   ] }</pre></div> <div>Parameter content type: <input type="text" value="application/json"/></div>	Schedule object to modify	body	<div>Model   <a href="#">Model Schema</a></div> <div><b>Schedule {</b> <b>id</b> (<a href="#">integer</a>): Unique identifier for the schedule, <b>name</b> (<a href="#">string</a>): Name of the schedule. May be localized or not (see implementation notes above), <b>intervals</b> (<a href="#">array[TimeWindow]</a>): Collection of the <b>TimeWindow</b>, <b>holidays</b> (<a href="#">array[integer]</a>, optional): List of holidays assigned to the schedule <b>}</b> <b>TimeWindow {</b> <b>daysOfTheWeek</b> (<a href="#">array[integer]</a>, optional): The days of the week on which to grant access, <b>start</b> (<a href="#">string</a>): The time during which the schedule will be active, <b>end</b> (<a href="#">string</a>): The time during which the schedule will be active <b>}</b></div>

Try it out!

[Hide Response](#)

Request URL

https://10.66.114.113:443/api/schedules/8

Response Body

no content

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:56:02 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "0",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]



# schedules : Operations about schedules

Show/Hide

List Operations

Expand Operations

Raw

DELETE

/schedules/{id}

Delete an schedule by id

## Implementation Notes

Permanently removes an schedule from the system, specified by the **id** in the url.

[List of response messages.](#)

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="8"/>	Unique identifier of the schedule to delete	path	integer

Try it out!

Hide Response

## Request URL

https://10.66.114.113:443/api/schedules/8

## Response Body

no content

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 00:57:54 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "0",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

# holidays : Operations about holidays

Show/Hide

List Operations

Expand Operations

Raw

GET

/holidays

Get all holidays

## Implementation Notes

Returns a list of **Holiday** from the TruPortal system, ordered by id.

This service can only return up to **250** of the holidays at one time. The **start** parameter can be used to specify the id of the holiday to begin with, if the parameter is not specified the first id (the smallest value) will be used. The **limit** parameter can be used to specify the number of holidays to be returned from the service call, only values between **1** and **250** are allowed.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Holiday** {  
  **id** (*integer*): Unique identifier for the holiday,  
  **name** (*string*): Name of the holiday  
}

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text"/>	Number of holidays to return (maximum 250)	query	integer
start	<input type="text"/>	Holiday id to start the results with	query	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/holidays

## Response Body

```
[
  {
    "id": 1,
    "name": "Holiday 1"
  },
  {
    "id": 2,
    "name": "Holiday 2"
  },
]
```

```
{
  "id": 3,
  "name": "Holiday 3"
},
{
  "id": 4,
  "name": "Holiday 4"
},
{
  "id": 5,
  "name": "Holiday 5"
},
{
  "id": 6,
  "name": "Holiday 6"
},
{
  "id": 7,
  "name": "Holiday 7"
},
}
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:00:40 GMT",
  "Content-Type": "application/json",
  "Content-Length": "258",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# holidays : Operations about holidays

Show/Hide

List Operations

Expand Operations

Raw

GET

/holidays/{id}

Get a holiday by id

## Implementation Notes

Returns a single **Holiday**, specified by the id in the url.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Holiday** {  
  **id** ([integer](#)): Unique identifier for the holiday,  
  **name** ([string](#)): Name of the holiday  
}

Response Content Type 

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<div>1</div>	Unique identifier of the holiday to return	path	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/holidays/1

## Response Body

```
{
  "id": 1,
  "name": "Holiday 1"
}
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:03:01 GMT",
  "Content-Type": "application/json",
  "Content-Length": "31",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Options": "nosniff",
}
```

```
"Cache-Control": "no-cache, no-store, private",  
"X-Xss-Protection": "1; mode=block"  
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

## userDefinedFields : Operations about user defined fields

GET /userDefinedFields

Show/Hide | List Operations | Expand Operations | Raw

Get all user-defined fields

### Implementation Notes

Returns a list of **UserDefinedField** from the TruPortal system, ordered by id.

This service can only return up to **250** of the user-defined fields at one time. The **start** parameter can be used to specify the id of the user-defined field to begin with, if the parameter is not specified the first id (the smallest value) will be used. The **limit** parameter can be used to specify the number of user-defined fields to be returned from the service call, only values between **1** and **250** are allowed.

The resulting user-defined fields may contain a localized string in the **name** field (only factory defined, not overridden fields are localized). The locale used for this field can be changed from the default of **en\_US** by setting the **locale** query parameter.

[List of response messages.](#)

### Response Class

Model | Model Schema

#### UserDefinedField {

- id** ([integer](#)): Unique identifier for the user-defined field,
- name** ([string](#)): Name of the user-defined field. May be localized or not (see implementation notes above),
- required** ([boolean](#)): Information if the user-defined field is required and cannot be empty,
- secured** ([boolean](#)): If set to True the user-defined field is visible only for users with special permission

}

Response Content Type 

application/json

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<div>3</div>	Number of user-defined fields to return (maximum 250)	query	integer
start	<div></div>	User-defined field id to start the results with	query	integer
locale	<div></div>	Locale used to resolve the names (see <a href="#">list of possible values</a> )	query	string

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/userDefinedFields?limit=3

## Response Body

```
[
  {
    "secured": false,
    "required": false,
    "id": 1,
    "name": "Person ID"
  },
  {
    "secured": false,
    "required": false,
    "id": 2,
    "name": "Title"
  },
  {
    "secured": false,
    "required": false,
    "id": 3,
    "name": "Department"
  }
]
```

## Response Code

```
200
```

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:07:49 GMT",
  "Content-Type": "application/json",
  "Content-Length": "204",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

## userDefinedFields : Operations about user defined fields

Show/Hide | List Operations | Expand Operations | Raw

GET

/userDefinedFields/{id}

Get a user-defined field by id

### Implementation Notes

Returns a single **UserDefinedField**, specified by the id in the url.

The resulting user-defined fields may contain a localized string in the **name** field (only factory defined, not overridden fields are localized). The locale used for this field can be changed from the default of **en\_US** by setting the **locale** query parameter.

[List of response messages.](#)

### Response Class

Model | Model Schema

**UserDefinedField {**

- id** (*integer*): Unique identifier for the user-defined field,
- name** (*string*): Name of the user-defined field. May be localized or not (see implementation notes above),
- required** (*boolean*): Information if the user-defined field is required and cannot be empty,
- secured** (*boolean*): If set to True the user-defined field is visible only for users with special permission

**}**

Response Content Type 

application/json

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<div>2</div>	Unique identifier of the user-defined field to return	path	integer
locale	<div></div>	Locale used to resolve the names (see <a href="#">list of possible values</a> )	query	string

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/userDefinedFields/2

### Response Body



```
{  
  "secured": false,  
  "required": false,  
  "id": 2,  
  "name": "Title"  
}
```

## Response Code

200

## Response Headers

```
{  
  "Server": "",  
  "Date": "Fri, 22 Oct 2010 01:10:03 GMT",  
  "Content-Type": "application/json",  
  "Content-Length": "63",  
  "Connection": "keep-alive",  
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",  
  "X-Content-Type-Options": "nosniff",  
  "Cache-Control": "no-cache, no-store, private",  
  "X-Xss-Protection": "1; mode=block"  
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

## actionTriggers : Operations about action triggers

Show/Hide

List Operations

Expand Operations

Raw

GET

/actionTriggers

Get all action triggers

### Implementation Notes

Returns a list of **ActionTrigger** from the TruPortal system, ordered by id.

This service can only return up to **250** of the action triggers at one time. The **start** parameter can be used to specify the id of the action trigger to begin with, if the parameter is not specified the first id (the smallest value) will be used. The **limit** parameter can be used to specify the number of action triggers to be returned from the service call, only values between **1** and **250** are allowed.

[List of response messages.](#)

### Response Class

Model

Model Schema

**ActionTrigger {**  
  **id** (*integer*): Unique identifier for the action trigger,  
  **name** (*string*): Name of the action trigger  
**}**

Response Content Type 

application/json

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text"/>	Number of action triggers to return (maximum 250)	query	integer
start	<input type="text"/>	Action trigger id to start the results with	query	integer

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/actionTriggers

### Response Body

```
[
  {
    "id": 1,
    "name": "Action Trigger 1"
  },
  {
    "id": 3,
    "name": "Action Trigger 3"
  }
]
```

```
},
{
  "id": 4,
  "name": "Action Trigger 4"
}
]
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:13:53 GMT",
  "Content-Type": "application/json",
  "Content-Length": "117",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# actionTriggers : Operations about action triggers

GET

/actionTriggers/{id}

Get an action trigger by id

Show/Hide | List Operations | Expand Operations | Raw

## Implementation Notes

Returns a single **ActionTrigger**, specified by the id in the url.

[List of response messages.](#)

## Response Class

Model [Model Schema](#)

**ActionTrigger {**  
  **id** ([integer](#)): Unique identifier for the action trigger,  
  **name** ([string](#)): Name of the action trigger  
**}**

Response Content Type 

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<div>1</div>	<b>Unique identifier of the action trigger to return</b>	path	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/actionTriggers/1

## Response Body

```
{
  "id": 1,
  "name": "Action Trigger 1"
}
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:16:34 GMT",
  "Content-Type": "application/json",
  "Content-Length": "37",
}
```

```
"Connection": "keep-alive",  
"Strict-Transport-Security": "max-age=31536000; includeSubDomains",  
"X-Content-Type-Options": "nosniff",  
"Cache-Control": "no-cache, no-store, private",  
"X-Xss-Protection": "1; mode=block"  
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

## actionTriggers : Operations about action triggers

Show/Hide

List Operations

Expand Operations

Raw

POST

/actionTriggers/{id}/execute

Execute an action trigger action

### Implementation Notes

Execute an action of a single action trigger, specified by the id in the url.

The action parameter is required and used to determine the action to perform on the action trigger: **activation** for action trigger activation and **deactivation** for action trigger deactivation.

[List of response messages.](#)

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="1"/>	Unique identifier of the action trigger to execute	path	integer
action	<input type="text" value="activation"/>	Action to perform on the action trigger	query	string

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/actionTriggers/1/execute?action=activation

### Response Body

no content

### Response Code

200

### Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:18:05 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "0",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

## devices : Operations about devices

Show/Hide

List Operations

Expand Operations

Raw

GET

/devices

Get all devices

### Implementation Notes

Returns a list of **Device** in the TruPortal system.

This service can only return up to **250** of the system devices at one time. The **start** parameter can be used to specify the id of the device to begin with, if the parameter is not specified the first device will be used. The **limit** parameter can be used to specify the number of devices to be returned from the service call, only values between **1** and **250** are allowed.

[List of response messages.](#)

### Response Class

Model

Model Schema

**Device** {  
  **id** (*integer*): Unique identifier for the device,  
  **name** (*string*): Name of the device specified in the **id**,  
  **linkedCamera** (*string*): id of the camera associated with this device  
}

Response Content Type 

application/json

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<div>4</div>	Number of devices to return (maximum 250)	query	integer
start	<div></div>	Devices id to start the result with	query	integer

Try it out!

[Hide Response](#)

### Request URL

https://10.66.114.113:443/api/devices?limit=4

### Response Body

```
[
  {
    "linkedCamera": 0,
    "id": 1,
    "name": "Controller (1)"
  },
  {
    "linkedCamera": 0,
    "id": 2,
```

```
"name": "Video Devices"
},
{
  "linkedCamera": 0,
  "id": 3,
  "name": "Door Controller 8888"
},
{
  "linkedCamera": 0,
  "id": 4,
  "name": "Door 8888-1"
}
]
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:21:43 GMT",
  "Content-Type": "application/json",
  "Content-Length": "226",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]



# devices : Operations about devices

Show/Hide

List Operations

Expand Operations

Raw

GET

/devices/{id}

Get a device by id

## Implementation Notes

Returns a single **device**, specified by the id in the url.

The **id** has to be defined and valid identifier for requested devices.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Device {**

- id** (*integer*): Unique identifier for the device,
- name** (*string*): Name of the device specified in the **id**,
- linkedCamera** (*string*): id of the camera associated with this device

**}**

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="3"/>	Unique identifier of the device to return	path	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/devices/3

## Response Body

```
{
  "linkedCamera": 0,
  "id": 3,
  "name": "Door Controller 8888"
}
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:23:53 GMT",
}
```

```
"Content-Type": "application/json",  
"Content-Length": "60",  
"Connection": "keep-alive",  
"Strict-Transport-Security": "max-age=31536000; includeSubDomains",  
"X-Content-Type-Options": "nosniff",  
"Cache-Control": "no-cache, no-store, private",  
"X-Xss-Protection": "1; mode=block"  
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# doors : Operations about doors

Show/Hide

List Operations

Expand Operations

Raw

GET

/devices/doors

Get all doors

## Implementation Notes

Returns a list of **Doors** in the TruPortal system.

This service can only return up to **250** of the system doors at one time. The **start** parameter can be used to specify the id of the door to begin with, if the parameter is not specified the first door will be used. The **limit** parameter can be used to specify the number of doors to be returned from the service call, only values between **1** and **250** are allowed.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Door {**

- id** (*integer*): Unique identifier for the door,
- name** (*string*): Name of the door specified in the **id**,
- state** (*string*) = ['offline' or 'tamper' or 'forced-open' or 'held-open' or 'unlocked' or 'credential-required' or 'locked-out']: State of the door,
- readerTamper** (*boolean*): Reader circuit is tampered,
- rteTamper** (*boolean*): Request to exit circuit is tampered,
- auxTamper** (*boolean*): Auxiliary circuit is tampered,
- linkedCamera** (*string*): id of the camera associated with this door

**}**

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text"/>	Number of doors to return (maximum 250)	query	integer
start	<input type="text"/>	Doors id to start the results with	query	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/devices/doors

## Response Body

```
[
  {
    "linkedCamera": 0,
    "auxTamper": false,
```

```
"name": "Door 8888-1",
"rteTamper": false,
"state": "credential-required",
"readerTamper": false,
"id": 4
},
{
  "linkedCamera": 0,
  "auxTamper": false,
  "name": "Door 8888-2",
  "rteTamper": false,
  "state": "credential-required",
  "readerTamper": false,
  "id": 5
}
]
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:27:03 GMT",
  "Content-Type": "application/json",
  "Content-Length": "296",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# doors : Operations about doors

Show/Hide

List Operations

Expand Operations

Raw

GET

/devices/doors/{id}

Get a door by id

## Implementation Notes

Returns a single **Door**, specified by the id in the url.

The **id** has to be defined and valid identifier for requested doors.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Door** {

- id** (*integer*): Unique identifier for the door,
- name** (*string*): Name of the door specified in the **id**,
- state** (*string*) = ['offline' or 'tamper' or 'forced-open' or 'held-open' or 'unlocked' or 'credential-required' or 'locked-out']: State of the door,
- readerTamper** (*boolean*): Reader circuit is tampered,
- rteTamper** (*boolean*): Request to exit circuit is tampered,
- auxTamper** (*boolean*): Auxiliary circuit is tampered,
- linkedCamera** (*string*): id of the camera associated with this door

}

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="4"/>	Unique identifier of the door to return	path	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/devices/doors/4

## Response Body

```
{
  "linkedCamera": 0,
  "auxTamper": false,
  "name": "Door 8888-1",
  "rteTamper": false,
  "state": "credential-required",
  "readerTamper": false,
  "id": 4
}
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:28:49 GMT",
  "Content-Type": "application/json",
  "Content-Length": "146",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

## doors : Operations about doors

Show/Hide

List Operations

Expand Operations

Raw

POST

/devices/doors/{id}/state

Modify a door state

### Implementation Notes

Set **Door** state, specified by the id in the url.

The **command** parameter is required and accepts one of the following values:

- **grant-access** - grants access (unlocks the door) for the Person specified by the **personId** parameter. If the **personId** parameter is not specified, then access is granted for the owner of the card with **cardNumber**. If **personId** and **cardNumber** are not defined, then access is granted to the user currently logged into the REST API (in the **/auth/login** service call). **NOTE:** Person's access permissions to the door are NOT checked.
- **open,unlock,secure,lockout,reinstate** - these commands are functionally equivalent to the relevant commands available in the main panel GUI. **personId** and **cardNumber** parameters are ignored. No specific Person/Credential is shown in the resulting log events.

[List of response messages.](#)

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="4"/>	Unique identifier of the door	path	integer
command	<input type="text" value="unlock"/>	State to set for the door	query	string
personId	<input type="text"/>	Unique identifier of the person being granted access	query	integer
cardNumber	<input type="text"/>	Card number of the credential being granted access	query	string

Try it out!

[Hide Response](#)

### Request URL

```
https://10.66.114.113:443/api/devices/doors/4/state?command=unlock
```

### Response Body

```
no content
```

### Response Code

```
200
```

### Response Headers

```
{
```

```
"Server": "",
"Date": "Fri, 22 Oct 2010 01:30:27 GMT",
"Content-Type": "text/html; charset=UTF-8",
"Content-Length": "0",
"Connection": "keep-alive",
"Strict-Transport-Security": "max-age=31536000; includeSubDomains",
"X-Content-Type-Options": "nosniff",
"Cache-Control": "no-cache, no-store, private",
"X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]



# readers : Operations about readers

Show/Hide

List Operations

Expand Operations

Raw

GET

/devices/readers

Get all readers

## Implementation Notes

Returns a list of **Reader** in the TruPortal system.

This service can only return up to **250** of the system readers at one time. The **start** parameter can be used to specify the id of the reader to begin with, if the parameter is not specified the first reader will be used. The **limit** parameter can be used to specify the number of readers to be returned from the service call, only values between **1** and **250** are allowed.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Reader {**

- id** (*integer*): Unique identifier for the reader,
- name** (*string*): Name of the reader specified in the **id**,
- doorId** (*integer*): Unique identifier of the door,
- direction** (*string*, optional) = ['entry' or 'exit']: Direction of the reader,
- linkedCamera** (*string*): id of the camera associated with this reader

**}**

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text" value="2"/>	Number of readers to return (maximum 250)	query	integer
start	<input type="text"/>	Readers id to start the result with	query	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/devices/readers?limit=2

## Response Body

```
[
  {
    "doorId": 4,
    "direction": "entry",
    "linkedCamera": 0,
    "id": 6,
```

```
    "name": "Reader 8888-1-1"
  },
  {
    "doorId": 4,
    "direction": "exit",
    "linkedCamera": 0,
    "id": 7,
    "name": "Reader 8888-1-2"
  }
]
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:33:35 GMT",
  "Content-Type": "application/json",
  "Content-Length": "183",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# readers : Operations about readers

Show/Hide

List Operations

Expand Operations

Raw

GET

/devices/readers/{id}

Get a reader by id

## Implementation Notes

Returns a single **Reader**, specified by the id in the url.

The **id** has to be defined and valid identifier for requested readers.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Reader** {

- id** (*integer*): Unique identifier for the reader,
- name** (*string*): Name of the reader specified in the **id**,
- doorId** (*integer*): Unique identifier of the door,
- direction** (*string*, optional) = ['entry' or 'exit']: Direction of the reader,
- linkedCamera** (*string*): id of the camera associated with this reader

}

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	6	Unique identifier of the reader to return	path	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/devices/readers/6

## Response Body

```
{
  "doorId": 4,
  "direction": "entry",
  "linkedCamera": 0,
  "id": 6,
  "name": "Reader 8888-1-1"
}
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:35:18 GMT",
  "Content-Type": "application/json",
  "Content-Length": "90",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# readers : Operations about readers

Show/Hide

List Operations

Expand Operations

Raw

GET

/devices/readers/groups

Get all reader groups

## Implementation Notes

Returns a list of **ReaderGroup** in the TruPortal system.

This service can only return up to **250** of the system reader groups at one time. The **start** parameter can be used to specify the id of the reader group to begin with, if the parameter is not specified the first reader group will be used. The **limit** parameter can be used to specify the number of reader groups to be returned from the service call, only values between **1** and **250** are allowed.

[List of response messages.](#)

## Response Class

Model

Model Schema

**ReaderGroup {**  
  **id** (*integer*): Unique identifier for the reader group,  
  **name** (*string*): Name of the reader group specified in the **id**  
**}**

Response Content Type 

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text"/>	Number of reader groups to return (maximum 250)	query	integer
start	<input type="text"/>	Reader groups id to start the result with	query	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/devices/readers/groups

## Response Body

```
[
  {
    "id": 1,
    "name": "Reader group 1"
  },
  {
    "id": 3,
    "name": "Reader group 3"
  },
]
```

```
{
  "id": 4,
  "name": "Reader group 4"
},
{
  "id": 5,
  "name": "Reader group 5"
}
]
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:37:21 GMT",
  "Content-Type": "application/json",
  "Content-Length": "148",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# readers : Operations about readers

Show/Hide

List Operations

Expand Operations

Raw

GET

/devices/readers/groups/{id}

Get a reader group by id

## Implementation Notes

Returns a single **ReaderGroup**, specified by the id in the url.

The **id** has to be defined and valid identifier for requested reader groups.

[List of response messages.](#)

## Response Class

Model

Model Schema

**ReaderGroup {**

- id** (*integer*): Unique identifier for the reader group,
- name** (*string*): Name of the reader group specified in the **id**

**}**

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	1	Unique identifier of the reader group to return	path	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/devices/readers/groups/1

## Response Body

```
{
  "id": 1,
  "name": "Reader group 1"
}
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:39:51 GMT",
  "Content-Type": "application/json",
  "Content-Length": "35",
}
```

```
"Connection": "keep-alive",  
"Strict-Transport-Security": "max-age=31536000; includeSubDomains",  
"X-Content-Type-Options": "nosniff",  
"Cache-Control": "no-cache, no-store, private",  
"X-Xss-Protection": "1; mode=block"  
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]



# inputs : Operations about inputs

Show/Hide

List Operations

Expand Operations

Raw

GET

/devices/inputs

Get all inputs

## Implementation Notes

Returns a list of **Input** in the TruPortal system.

This service can only return up to **250** of the system inputs at one time. The **start** parameter can be used to specify the id of the input to begin with, if the parameter is not specified the first input will be used. The **limit** parameter can be used to specify the number of inputs to be returned from the service call, only values between **1** and **250** are allowed.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Input** {  
  **id** (*integer*): Unique identifier for the input,  
  **name** (*string*): Name of the input specified in the **id**,  
  **state** (*string*) = ['offline' or 'active' or 'inactive' or 'tamper']: State of the input,  
  **linkedCamera** (*string*): id of the camera associated with this input  
}

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text"/>	Number of inputs to return (maximum 250)	query	integer
start	<input type="text"/>	Inputs id to start the result with	query	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/devices/inputs

## Response Body

```
[
  {
    "state": "inactive",
    "linkedCamera": 0,
    "id": 11,
    "name": "Input (11)"
  },
  {
```

```
"state": "inactive",
"linkedCamera": 0,
"id": 12,
"name": "Input (12)"
},
{
"state": "inactive",
"linkedCamera": 0,
"id": 13,
"name": "Input (13)"
},
{
"state": "inactive",
"linkedCamera": 0,
"id": 14,
"name": "Input (14)"
}
]
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:42:49 GMT",
  "Content-Type": "application/json",
  "Content-Length": "296",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# inputs : Operations about inputs

Show/Hide

List Operations

Expand Operations

Raw

GET

/devices/inputs/{id}

Get an input by id

## Implementation Notes

Returns a single **Input**, specified by the id in the url.

The **id** has to be defined and valid identifier for requested inputs.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Input {**

- id** (**integer**): Unique identifier for the input,
- name** (**string**): Name of the input specified in the **id**,
- state** (**string**) = ['offline' or 'active' or 'inactive' or 'tamper']: State of the input,
- linkedCamera** (**string**): id of the camera associated with this input

**}**

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	11	Unique identifier of the item to return	path	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/devices/inputs/11

## Response Body

```
{
  "state": "inactive",
  "linkedCamera": 0,
  "id": 11,
  "name": "Input (11)"
}
```

## Response Code

200

## Response Headers

```
{
```

```
"Server": "",  
"Date": "Fri, 22 Oct 2010 01:45:23 GMT",  
"Content-Type": "application/json",  
"Content-Length": "72",  
"Connection": "keep-alive",  
"Strict-Transport-Security": "max-age=31536000; includeSubDomains",  
"X-Content-Type-Options": "nosniff",  
"Cache-Control": "no-cache, no-store, private",  
"X-Xss-Protection": "1; mode=block"  
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

# outputs : Operations about outputs

Show/Hide

List Operations

Expand Operations

Raw

GET

/devices/outputs

Get all outputs

## Implementation Notes

Returns a list of **Output** in the TruPortal system.

This service can only return up to **250** of the system outputs at one time. The **start** parameter can be used to specify the id of the output to begin with, if the parameter is not specified the first output will be used. The **limit** parameter can be used to specify the number of outputs to be returned from the service call, only values between **1** and **250** are allowed.

[List of response messages.](#)

## Response Class

Model

Model Schema

**Output {**

- id** (*integer*): Unique identifier for the output,
- name** (*string*): Name of the output specified in the **id**,
- state** (*string*) = ['offline' or 'active' or 'inactive']: State of the output,
- linkedCamera** (*string*): id of the camera associated with this output

**}**

Response Content Type

application/json

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
limit	<input type="text"/>	Number of outputs to return (maximum 250)	query	integer
start	<input type="text"/>	Outputs id to start the result with	query	integer

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/devices/outputs

## Response Body

```
[
  {
    "state": "active",
    "linkedCamera": 0,
    "id": 15,
    "name": "Output (15)"
  },
]
```

```
{
  "state": "active",
  "linkedCamera": 0,
  "id": 16,
  "name": "Output (16)"
}
```

```
]
```

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:48:38 GMT",
  "Content-Type": "application/json",
  "Content-Length": "146",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]

## outputs : Operations about outputs

[Show/Hide](#)[List Operations](#)[Expand Operations](#)[Raw](#)

GET

/devices/outputs/{id}

[Get an output by id](#)

### Implementation Notes

Returns a single **Output**, specified by the id in the url.

The **id** has to be defined and valid identifier for requested outputs.

[List of response messages.](#)

### Response Class

[Model](#) | [Model Schema](#)

#### Output {

**id** ([integer](#)): Unique identifier for the output,  
**name** ([string](#)): Name of the output specified in the **id**,  
**state** ([string](#)) = ['offline' or 'active' or 'inactive']: State of the output,  
**linkedCamera** ([string](#)): id of the camera associated with this output  
}

Response Content Type `application/json`

### Parameters

Parameter	Value	Description	Parameter Type	Data Type
<b>id</b>	<b>15</b>	<b>Unique identifier of the output to return</b>	path	integer

[Try it out!](#)[Hide Response](#)

### Request URL

`https://10.66.114.113:443/api/devices/outputs/15`

### Response Body

```
{
  "state": "active",
  "linkedCamera": 0,
  "id": 15,
  "name": "Output (15)"
}
```

### Response Code

`200`

### Response Headers

```
{
```

```
"Server": "",  
"Date": "Fri, 22 Oct 2010 01:50:23 GMT",  
"Content-Type": "application/json",  
"Content-Length": "71",  
"Connection": "keep-alive",  
"Strict-Transport-Security": "max-age=31536000; includeSubDomains",  
"X-Content-Type-Options": "nosniff",  
"Cache-Control": "no-cache, no-store, private",  
"X-Xss-Protection": "1; mode=block"  
}
```

[ BASE URL: <https://10.66.114.113/api-docs> , API VERSION: 1.0.0 ]



# outputs : Operations about outputs

Show/Hide

List Operations

Expand Operations

Raw

POST

/devices/outputs/{id}/state

Modify an output state

Implementation Notes

Modify an output state.

[List of response messages.](#)

## Parameters

Parameter	Value	Description	Parameter Type	Data Type
id	<input type="text" value="15"/>	Unique identifier of the output	path	integer
state	<input type="text" value="inactive"/>	State to set for the output	query	string

Try it out!

[Hide Response](#)

## Request URL

https://10.66.114.113:443/api/devices/outputs/15/state?state=inactive

## Response Body

no content

## Response Code

200

## Response Headers

```
{
  "Server": "",
  "Date": "Fri, 22 Oct 2010 01:52:04 GMT",
  "Content-Type": "text/html; charset=UTF-8",
  "Content-Length": "0",
  "Connection": "keep-alive",
  "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
  "X-Content-Type-Options": "nosniff",
  "Cache-Control": "no-cache, no-store, private",
  "X-Xss-Protection": "1; mode=block"
}
```

# TruPortal REST API Documentation

This page contains additional information about the API response messages and error codes, which are common to all the functions. A full list of event types is also provided in this reference.

## Response messages

### Response status codes

The following error/status codes may be returned as a result of an API function call. The **response body** may contain additional information, or may be empty depending on the **status code**.

HTTP Status Code	Description	Response body
200	Success. The API function has been executed properly.	The response body is API-related. It can be empty (e.g. object deletion), or it may contain a single integer value (e.g. ID of newly created object), JSON data, binary data (e.g. picture). Please refer to the documentation of the particular API function call for detailed information.
204	Item not found. Unable to execute API function on the item because it does not exist.	Empty body.
400	Bad request response. Bad request can be related to invalid API function parameters, required JSON property is not found, etc.	Error description body - <a href="#">see below</a> .
401	Invalid session/unauthorized. Session key has expired or invalid credentials provided when logging into the API.	Error description body - <a href="#">see below</a> .
403	Invalid permissions. API function cannot be performed due to invalid permissions.	Error description body - <a href="#">see below</a> .
404	Invalid URL/path. Could not find API function for specified URL.	Error description body - <a href="#">see below</a> .
405	Invalid method. URL is valid but could not find API function for specified method.	Error description body - <a href="#">see below</a> .
500	Internal error - unable to execute API function properly. Invalid request body format (e.g. improperly formatted JSON) or internal subcomponent fault.	Error description body - <a href="#">see below</a> .

### Error description message body

Most error messages are returned formatted with the following JSON structure:

```
{
```

```
"code": "" (short text describing the meaning of HTTP status code)
"message": "" (more detailed API-related information about the failure)
}
```

If the status **code** returned is **500**, the error may contain additional information within the **message** field. Here is an example of an Internal error message:

```
{
  "code": "500 Internal Server Error"
  "message": "[S/E]: SOAP-ENV:Client: Webservices Fault."
}
```

The **message** displayed here contains two additional numeric values in square brackets that refer to the subsystem number (**S**) and error code (**E**) of the internal error.

## Subsystem number

Some error messages are accompanied by a **subsystem** number. This number indicates the source of the error/fault.

A list of the possible values are displayed below:

Subsystem number	Subsystem description
1	SQL Subsystem
2	Access Control System
3	Socket Layer
4	Request Subsystem
5	Response Thread
6	Message Subsystem
7	reserved
8	Linux Error
9	Web Services
10	Event Subsystem
11	reserved
12	reserved
13	SP14 API Subsystem

## Error codes

Error codes are divided into three types: [General errors](#), [SQL errors](#) and [SP14 API errors](#).

A list of the possible values are displayed below:

General errors

Error code	Error description
1000	Insufficient memory
1001	reserved
1002	Message on queue too large
1003	Null pointer
1004	Card number greater than 32 characters
1005	PIN greater than 16 characters
1006	Unimplemented call
1007	Invalid schedule
1008	Null pointer
1009	Credential ID already exists
1010	Feature not supported
1011	No such record
1012	Record already exists
1013	Record in use
1014	Credential ID does not exist
1015	File write failure
1016	File open failure
1017	Restart task failure
1018	Empty data record
1019	Record does not exist
1020	Events update in progress
1021	Invalid input data
1022	reserved
1023	Error sending communication message
1024	Feature not yet implemented
1025	Critical panel data out of sync
1026	No serial free node
1027	Random number generator failure
1028	Write access locked
1029	Invalid session

1030	Invalid database data
1031	Door disabled
1032	Duplicate login name
1033	Cannot delete admin user
1034	Data limit reached
1035	Restore file not found
1036	Error getting checksum from file name
1037	Failed to calculate file checksum
1038	File checksum does not match
1039	Tar validation failed to compare
1040	Restore database controller invalid
1041	Logical software error
1042	Invalid card format
1043	Invalid query
1044	Card format in use
1045	Error creating safe string
1046	Identical card format exists
1047	Input error - password empty
1048	Input error - username empty
1049	Username and/or password not found
1050	Access denied, invalid permissions
1051	Invalid database request
1052	Database write lock failed
1053	Invalid database transaction
1054	Database table full
1055	Login name already in use
1056	reserved
1057	NTP sync failed
1058	Incorrect IO parent ID
1059	Network information read failure
1060	Network information write failure
1061	System information read failure
1062	All doors command failure
1063	Cannot add built-in controller I/O

1064	Object requires resync
1065	Database handle invalid
1066	Certificate generation error
1067	Invalid password
1068	Photo file too large
1069	Photo file write error
1070	Failed to resize photo
1071	Photo file too large after resize
1072	Language pack file not found
1073	Language pack file is invalid
1074	File version is not compatible with firmware
1075	Language pack install general error
1076	Language is no longer installed on the system.
1077	Email script error
1078	Invalid test email
1080	File decryption failed
1081	File unzip failed
1082	Required file is missing
1083	Locale code not found
1084	Install failed - locale file system full
1085	Specified port is in use
1100	Action Trigger - too many triggers
1101	Action Trigger - too many queries
1102	Action Trigger - disabled
1105	Action Trigger - invalid entity ID
1106	Action Trigger - invalid query type
1107	Action Trigger - invalid command type
1108	Action Trigger - invalid trigger type
1109	Action Trigger - invalid query match
1110	Action Trigger - invalid query scope
1111	Action Trigger - invalid command scope
1112	Action Trigger - invalid command point
1113	Action Trigger - out of bytecode
1114	Action Trigger - out of command points

1115	Action Trigger - out of outputs
1120	Scheduled backup was successful, but file transfer failed.
1121	Attempt to update the cron job scheduler has failed.
1122	Scheduled jobs must be at least 30 minutes apart.
1123	Invalid scheduled job type
1124	The scheduled backup failed
1125	Cannot delete network share default
1126	Attempting to enable a scheduled job with the network share of none is not allowed.
1127	Default network share record cannot be modified.
1128	Network share records cannot be modified to protocol none.
1129	A scheduled job is currently running.
1133	Attempt to change system language to a language that has been removed.
1134	Invalid brand detected in firmware update file.
1136	Firmware upgrade failed. Invalid device type.
1137	Firmware upgrade failed. General error.
1138	Firmware upgrade failed. Device is offline.
1200	A general curl error has occurred. Please check CURL documentation for specific details for error code 12xx, where XX is the CURL error code.
1203	URL not properly formatted.
1204	Feature, protocol, or option not built in.
1205	Could not resolve proxy.
1206	Could not resolve host.
1207	Failed to connect to host or proxy.
1209	Remote access denied.
1210	FTP error.
1235	SSL error.
1245	Outgoing connection Interface cannot be used.
1247	Too many redirects.
1252	Nothing was returned from the server.
1255	Send/Receive network data error.
1267	Remote server denied curl to login.
1300	Invalid arguments
1301	Send email failed
1302	Email notifications not enabled
1303	No sender email

1304	No email recipient
------	--------------------

## SQL errors

Error code	Error description
0	Successful result
1	SQL error or missing database
2	Internal logic error in SQLite
3	Access permission denied
4	Callback routine requested an abort
5	The database file is locked
6	A table in the database is locked
7	A malloc() failed
8	Attempt to write a read-only database
9	Operation terminated by sqlite3_interrupt()
10	Some kind of disk I/O error occurred
11	The database disk image is malformed
12	NOT USED. Table or record not found
13	Insertion failed because database is full
14	Unable to open the database file
15	Database lock protocol error
16	Database is empty
17	The database schema changed
18	String or BLOB exceeds size limit
19	Abort due to constraint violation
20	Data type mismatch
21	Library used incorrectly
22	Uses OS features not supported on host
23	Authorization denied
24	Auxiliary database format error
25	2nd parameter to sqlite3_bind out of range
26	File opened that is not a database file
100	sqlite3_step() has another row ready
101	sqlite3_step() has finished executing

## SP14 API errors



Error code	Error description
-2147483648	IPC error start
-2147483647	Timeout waiting for core
-2147483646	Unknown reply received
-2147483645	Datagram is smaller than the header size
-2147483644	Datagram size doesn't match header->dataSize
-2147483643	Unknown header->messageID value
-2147483642	Data size incorrect for messageID
-2147483641	IPC Call is not currently implemented.
-2147483640	Too many logs requested
-2147483639	Database error
-2147483638	Authentication of host failed

## Event types with descriptions

### Event types

The **event type** is a numeric value which identifies each type of event. The table below shows all of the possible event types, along with a short description. When logged into the TruPortal system from a browser, these **event types** can also be found in parentheses after each item in the event log.

The events returned by TruPortal may be requested in multiple languages. For a list of supported languages, refer to the [locales section](#) in this document.

Please note that not all of the events in the following table may be reported by a TruPortal system. Some event types may not be used by TruPortal, or may only be used for internal purposes.

Some events may contain additional information such as **device name**, **person name**, or **credential id**.

Event type	Event description
33	Cabinet Tamper
4096	Upload Okay
4097	Bad System ID
4098	System Password Fail
4099	Config No Reply
4100	Program Lost
4101	Spring Forward
4102	Fall Back
4105	Device Communications Fail
4106	Device Communications Restored

4107	Tamper Alarm
4108	Time Lost
4109	Year Change
4112	Anti-Passback Reset - Person
4123	Exit Push Button Fail
4128	Door Alarm - Held Open
4129	Door Restored
4130	Door Forced Alarm
4131	Door Trouble Open/Short Circuit
4132	Door Mag Trouble
4138	Access Denied - Duplicate Card
4139	Operator Access Override
4140	Door Held Open Restore
4141	Door Forced Restore
4142	Door Mag Trouble Restore
4145	Output Error
4161	Alarm Canceled
4162	Alarm Active
4164	reserved
4167	Action Trigger Device Offline
4170	Input Disabled
4171	Input Enabled
4172	Action Trigger Activated
4173	Action Trigger Deactivated
4224	Time Changed
4225	Bad Pin Lockout
4230	Anti-Passback Reset - Global
4231	Anti-Passback Reset - Area
4232	Bad Card Lockout
4233	All Lockouts Removed
4352	Card Assigned
4416	Reader in Card + PIN Mode
4480	Reader in Card Only Mode
4544	Reader in Card or Common PIN Mode

4608	Reader in PIN Only Mode
4672	Door Locked
4736	Door Unlocked
4800	Reader Locked Out
4864	Reader Reinstated
5056	Panic Token
5120	Reader Normal Mode
5248	Reader Escort Mode
5824	Door Momentarily Unlocked
5888	Pending First User
6080	Card Deleted
6144	Access Granted
6208	reserved
6272	reserved
6336	reserved
6400	reserved
6464	Access Granted - Entry Made
6528	Egress Granted
6720	Access Denied - Unauthorized
6848	Access Denied - Wrong Time
6912	Access Denied - Invalid Facility Code
6976	Access Denied - Invalid PIN
7040	Access Denied - Anti-Passback Violation
7104	Access Denied - Invalid Issue Code
7360	Access Denied - Time expired
7808	Access Granted - No Entry
7872	Access Granted - Pending Entry
8000	Access Denied - Door Locked Out
10240	Output On
11264	Output Off
13376	Access Denied - User No Access
13632	Door Held Open Pre Alarm
13696	Reader in Facility Only Mode
14016	Access Granted - Soft Anti-Passback Violation

14080	Access Granted - Soft Anti-Passback Violation with No Entry
14336	Egress Granted - Soft Anti-Passback Violation with No Entry
14400	Egress Granted - Soft Anti-Passback Violation
14591	External event generated via the REST API
14592	reserved
14593	reserved
14594	reserved
14595	Time was synced to time server
14596	reserved
14597	reserved
14598	reserved
14599	reserved
14600	reserved
14601	reserved
14602	reserved
14610	Panel Reboot to Factory Defaults
14611	Panel Restart Due to Power Failure
14612	Backup Battery Critical
14613	Backup Battery Cutoff
14614	USB 1 Fuse Blown
14615	USB 2 Fuse Blown
14616	USB 1 Fuse Reset
14617	USB 2 Fuse Reset
14618	Memory Backup Battery Low
14619	Panel Reset Watchdog
14620	NTP Time Synced
14621	NTP Time Sync failed
14622	System Trouble
14623	System Restored
14624	Backup Battery Low
14625	Backup Battery Restored
14626	AC Power Failed
14627	AC Power Restored
14628	Device Failed

14629	Device Restored
14630	Device Battery Failed
14631	Device Battery Restored
14632	Door Tamper Alarm
14633	Door Tamper Restored
14640	Input Active
14641	Input Tamper Alarm
14642	Input Inactive
14643	Tamper Restored
14644	All Doors Unlocked
14645	All Doors Reinstated
14646	All Doors Locked Out
14647	Time Changed From
14648	Time Changed To
14649	Backup Battery Not Detected
14650	User Login
14651	Fuse Tripped
14652	Fuse Restored
14653	Scheduled DB Backup Successful
14654	Scheduled DB Backup Failed
14655	Email Sent Successfully
14656	Email Send Failed
14657	Email Send Expired
14658	Email Send Disabled
14659	Scheduled Event Backup Successful
14660	Scheduled Event Backup Failed
14661	Action Trigger Manually Activated
14662	Action Trigger Manually Deactivated
16384	Access Denied - Invalid Credential
32768	Access Denied - Duplicate Card
49152	Bad Card Format

## Locales

The following table contains the locale codes supported by the TruPortal system. The codes may be passed as the **locale** parameter to the events and user-defined fields services.

If the locale requested is not currently installed on the TruPortal system, **English (US)** will be used instead.

Locale code	Language
de_DE	Deutsch (German)
en_GB	English (GB)
en_US	English (US, <b>default</b> )
es_ES	Español (Spanish)
fi_FI	Suomalainen (Finnish)
fr_FR	Français (French)
it_IT	Italiano (Italian)
n1_NL	Nederlands (Dutch)
pt_BR	Português (Portuguese)
sv_SE	Svensk (Swedish)
tr_TR	Türk (Turkish)
zh_CN	简体中文 (Chinese)

## End User License Agreement for TruPortal SDK

BY DOWNLOADING, INSTALLING OR USING THE SOFTWARE, YOU AGREE TO BE BOUND BY THIS END USER LICENSE AGREEMENT (THE “AGREEMENT”). IF YOU DO NOT AGREE, DO NOT INSTALL, COPY OR USE THE SOFTWARE; YOU MUST DELETE OR DESTROY ANY COPY IN YOUR POSSESSION OR CONTROL. This Agreement is between you (“you” or “your”) and UTC FIRE & SECURITY AMERICAS CORPORATION, INC. (“Licensor”) for the TRUPORTAL software that accompanies this Agreement, including all associated media, printed or electronic documentation, application programming interfaces (“APIs”), reference designs if provided or, if not, samples and internet-based services (collectively, the “Software”).

### 1. Subject Matter

- 1.1. *Software. Design, Development and Testing.* Licensor grants to you a personal, nonexclusive, nonsublicensable, nontransferable, revocable license to install and use the Software for the purposes of designing, developing and testing your software or product that operates with the Software; and to reproduce, use and modify the sample source code (“**Sample Code**”) for the purposes of designing, developing and testing your software products and distribute, in object code form, the Sample Code and your modifications thereto as part of your software/product that works with the TRUPORTAL Software.
- 1.2. *Documentation.* Licensor grants to you a personal, nonexclusive, nonsublicensable, nontransferable, revocable license to make and use a reasonable number of copies of documentation provided with the Software for your internal use in connection with your use of the Software; provided that you reproduce all copyright and other proprietary notices that are on the original copies.

### 2. License Limitations. The license(s) granted in Section 1 are conditioned upon your compliance with the following limitations:

- 2.1. *Reverse Engineering.* You may not decompile, decipher, disassemble, reverse engineer or otherwise attempt to access source code of the Software, or circumvent any technical limitations in the Software that limit or restrict access to or use of the Software or any content, file, or other work, except as expressly permitted by applicable law notwithstanding this limitation.
- 2.2. *No Distribution, Rental or Transfer.* You may not distribute (except as may be expressly permitted herein), publish, rent, lease, lend, transfer, sublicense, disclose or otherwise provide the Software to any third party.
- 2.3. *No Modification or Derivative Works.* Except as may be expressly permitted herein, You may not modify or create derivative works of the Software, in whole or in part.
- 2.4. *Proprietary Notices.* You may not remove any proprietary notices or labels on the Software or any copy thereof.
- 2.5. *Non-Permitted Uses.* Without limiting any of the foregoing, you may not make any use of the Software in any manner or for any purpose not expressly permitted by this Agreement.
- 2.6. *Distribution Requirements.* If you choose to exercise the licenses to redistribute Sample Code, any redistribution by you is subject to your compliance with this Section:
  - (i) you will only distribute the Sample Code in object code form and in conjunction with and as a part of a software product developed by you that adds significant and primary functionality to the Sample Code (“**Licensee Software**”);
  - (ii) you will ensure that the Redistributable Code only operates in conjunction with the TRUPORTAL Software in support of the intended purpose of the TRUPORTAL Software;
  - (iii) you will only distribute Licensee Software containing the Sample Code pursuant to an end user license agreement with terms no less protective than those contained in this Agreement;
  - (iv) you will not to use Licensor’s name, logo or trademarks;

- (v) the Licensee Software will display a valid copyright notice which will be sufficient to protect Licensor's copyright in the Software;
  - (vi) you will not permit further distribution of the Sample Code by your end users or licensees, *except* you may permit further redistribution of the Sample Code as part of, the Licensee Software;
  - (vii) you will not distribute the Sample Code in any manner that would cause the Sample Code to become subject to any of the terms of an Open Source License (defined below); and
  - (viii) you will not create derivative works of the Sample Code in any manner that would cause the Sample Code in whole or in part to become subject to any of the terms of an Open Source License. **"Open Source License"** means any license that requires as a condition of use, modification or distribution of software subject to the Open Source License, that the software or other software combined or distributed with the software be (x) disclosed or distributed in source code form; (y) licensed for the purpose of making derivative works; or (z) redistributable at no charge. "Open Source License" includes, without limitation, any GNU General Public License ("GPL").
3. **Reservation of Rights and Ownership.** Licensor, its parent or its suppliers own all right, title, and interest, including all intellectual property rights, in and to the Software and reserve all rights not expressly granted to you in this Agreement. The Software is protected by copyright and other intellectual property laws and treaties.
  4. **Additional Software or Services.** This Agreement applies to any updates, supplements, add-on components, and internet-based service components (if any) of the Software that Licensor may, in its sole discretion, provide or make available to you ("**Update**"). If Licensor provides additional terms along with the Update, those terms will apply to the Update. If Licensor provides you an Update, Licensor may, at its sole discretion, require you to use Update and cease use of early versions. Licensor reserves the right to discontinue any internet-based service (if any) provided or made available to you through the use of the Software.
  5. **Termination.** This Agreement will automatically terminate upon your breach of any of this Agreement's terms and conditions. In the event of termination, you must immediately destroy all copies of the Software, and the following Sections of this Agreement will survive: Sections 2, 3 and 5 to 11.
  6. **DISCLAIMER OF WARRANTIES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW THE SOFTWARE IS PROVIDED AS IS AND WITH ALL FAULTS. LICENSOR AND ITS PARENT, AFFILIATES AND SUPPLIERS HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING ANY IMPLIED WARRANTIES, DUTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT, LACK OF VIRUSES OR BUGS, ACCURACY OR COMPLETENESS OF RESPONSES OR RESULTS WITH REGARD TO THE SOFTWARE. LICENSOR AND ITS PARENT, AFFILIATES AND SUPPLIERS DO NOT WARRANT THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE OR THAT ANY DEFECTS WILL BE CORRECTED. THE ENTIRE RISK ARISING OUT OF THE SOFTWARE REMAINS WITH YOU.**
  7. **EXCLUSIONS OF CERTAIN DAMAGES; LIMITATIONS OF LIABILITY. IN NO EVENT WILL LICENSOR OR ITS PARENT, AFFILIATES OR SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, PUNITIVE, INDIRECT OR CONSEQUENTIAL DAMAGES (INCLUDING DAMAGES FOR LOSS OF PROFITS, DAMAGES TO ANY COMPUTER, DEVICE, OR SYSTEM, LOSS OF DATA, GOODWILL, USE OR OTHER LOSSES) ARISING OUT OF OR IN ANY WAY RELATED TO THE SOFTWARE OR THIS AGREEMENT, REGARDLESS OF THE CAUSE OF ACTION OR THE BASIS OF THE CLAIM AND EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF THE DAMAGES OR REMEDIES FAIL OF THEIR ESSENTIAL PURPOSE.**  
  
**LICENSOR'S AND ITS PARENT'S, AFFILIATES' AND SUPPLIERS' ENTIRE LIABILITY UNDER THIS AGREEMENT AND YOUR EXCLUSIVE REMEDY WILL BE LIMITED TO THE ACTUAL DAMAGES YOU INCUR IN REASONABLE RELIANCE ON THE SOFTWARE UP TO THE PRICE YOU PAID FOR THE SOFTWARE. NO ACTION, REGARDLESS OF FORM, RELATING TO THE SOFTWARE MAY BE BROUGHT BY YOU MORE THAN ONE YEAR AFTER YOU HAVE KNOWLEDGE OF THE OCCURRENCE WHICH GIVES RISE TO THE CAUSE OF ACTION.**



8. **Indemnification.** You agree to defend, indemnify and hold harmless Licensor and its parent, affiliates and suppliers and their respective officers, directors and employees from all claims and expenses (including attorneys' fees and costs) that arise out of or in connection with your use of the Software, use or distribution of the Licensee Software, any breach of this Agreement, or your violation of any laws or regulations or the rights of any third party.
9. **U.S. Government License Rights.** Software provided to the U.S. Government pursuant to solicitations issued on or after December 1, 1995 is provided with the commercial license rights and restrictions described in this Agreement. Software provided to the U.S. Government pursuant to solicitations issued prior to December 1, 1995 is provided with "Restricted Rights" as provided for in FAR, 48 CFR 52.227-14 (JUNE 1987) or DFAR, 48 CFR 252.227-7013 (OCT 1988), as applicable.
10. **Compliance with Law; Export Restrictions.** You will comply with all applicable international and national laws, rules and regulations that apply to the Software and your use of the Software, including the U.S. Export Administration Regulations, as well as end user, end use, and destination restrictions issued by U.S. or other governments. You acknowledge that the Software is of U.S. origin and subject to U.S. export jurisdiction.
11. **Governing Law and Jurisdiction.** This Agreement will be construed and controlled by New York law, without giving effect to its conflict of law provisions. Each party consents to exclusive jurisdiction and venue in the state and federal courts in New York for any and all disputes, claims and actions arising from or in connection with the Software and this Agreement. The United Nations Convention on Contracts for the International Sale of Goods does not apply to this Agreement.
12. **General.** The section titles in this Agreement are used solely for the parties' convenience and have no legal or contractual significance. Any list of examples following "including" or "e.g.," is illustrative and not exhaustive, unless qualified by terms like "only" or "solely." Licensor's failure to act with respect to a breach by you or others does not waive its right to act with respect to subsequent or similar breaches. No waiver of any provision of this Agreement will be effective unless it is in a signed writing, and no waiver will constitute a waiver of any other provision(s) or of the same provision on another occasion. If a court of competent jurisdiction holds any term, covenant or restriction of this Agreement to be illegal, invalid or unenforceable, in whole or in part, the remaining terms, covenants and restrictions will remain in full force and effect and will in no way be affected, impaired or invalidated. You may not assign, transfer or sublicense this Agreement or your rights (if any) under this Agreement. This Agreement will be binding upon all successors and assigns. This Agreement constitutes the entire agreement between you and Licensor with respect to the Software and merges all prior and contemporaneous communications and proposals, whether electronic, oral or written, between you and Licensor with respect to the Software. All notices to Licensor in connection with this Agreement must be in writing and will be deemed given as of the day they are deposited in the U.S. mails, postage prepaid, certified or registered, return receipt requested or sent by overnight courier, charges prepaid to the address set forth below.

## **Contact Information**

If you have any questions about this Agreement, or want to contact Licensor for any reason, please direct all correspondence to your Interlogix or Lenel sales representative.

[www.interlogix.com](http://www.interlogix.com)

Customer support: [www.interlogix.com/support](http://www.interlogix.com/support)

© 2017 United Technologies Corporation. All rights reserved.