**Booking Software & Building Controlling System**

**Data Exchange Protocol**

**Nordic Standard**

**Draft 1**

DocumentVersion 0.0.6

2016-09-27

**Companies that conforms to this standard/document**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Country** | **Date** | **Contact** | **Color** |
| Regler Och Webbteknik Sverige AB | Sweden | 2016-09-21 | Robin Andersson  [robin@rows.se](mailto:robin@rows.se) | Yellow |
| Agrando Sweden AB | Sweden | 2016-09-22 | Rune Hansen [rune.hansen@agrando.no](mailto:rune.hansen@agrando.no) | Blue |
| LJ System AB | Sweden | 2016-09-23 | Niklas Falemar  [niklas@ljsystem.se](mailto:niklas@ljsystem.se) | Orange |
| Eniac Data AB | Sweden | 2016-09-23 | David Fröjmark  [david.frojmark@eniac.se](mailto:david.frojmark@eniac.se) | Green |
| JEFF Electronics AB | Sweden | 2016-09-26 | Björn Granbom  [bjorn@jeff.se](mailto:bjorn@jeff.se) | Red |
| Mild Media AB | Swden | 2016-09-27 | Albin Nystedt | Purple |

Every change a company want to make to this document must be highlighted with corresponding color. When the change is unanimous and democratically decided to stay the color will be removed, and (very important) the fixed change must be noted/listed in the “Document version notes” section. Both of who asks and the one who approved the change must be noted.

**Introduction**

The goal with this standard is to simplify and unify the data exchanges between booking administration softwares and building controlling systems.

This standard is in its early stage and will be updated regularly in the coming year.

**Abbreviations and explanations**

**Booking administration software:** is referred to an administration software where a user has the ability to schedule the use of one or more resources.

In this document Booking Administration Software will be shorted to BAS.

**Building controlling systems:** is refereed to systems that in one way or another controls a building and/or its surroundings.

In this document Building Controlling Systems will be shorted to BCS.

**Data Transfer**

The data are to be transferred with HTTPS as a POST method.

The URL shall always be the same regardless of what method is invoked.

All transfers are initiated by the BCS. Thus all requests will be made by a BCS and all responses comes from a BAS.

HTTP Keep-Alive are preferred.

**Request/response payload**

The payload data must be formatted as valid **JSON**.

All variable names are **case sensitive**.

The order of variables is **not** **predetermined** and can be in any order within the JSON object.

All ID values must be valid GUID/UUID string.

Every request must include a variable called “appID” which contains a unique identification of the BCS system so that the BAS can filter unwelcome requests. “appID” works as login and password and shall not be reused between different BAS or BCM.

The JSON object must be passed as the body of the POST request.

TODO: A verification with some sort of hashing needs to be implemented.

**Fetching customer data**

*This method requests information about the customer and their resources.*

**REQUEST**

|  |  |  |
| --- | --- | --- |
| Name | Description | Note |
| method | Requested method. Every request must have this. |  |
| version | Request version number. This is to ensure that future changes that break backward compatibility does not destroy existing connections. |  |
| appId | A unique identification of the BCS system. Must be GUID/UUID compatible. Every single BCSs should be able to have the same appId between several BASs. |  |
| customers | A GUID/UUID list of requested customers. The customer id is determined in BAS and handed to BCS prior to installation. |  |

*example:*

{

“method”: “GetCustomerData”,

”version”: 1,

“appId”: “4e481c08-c808-4188-b128-20a5a02ebf96”,

"customers": ["e3941203-37c8-4aaf-a10c-a46100ccb787"]

}

**RESPONSE**

|  |  |  |
| --- | --- | --- |
| Name | Description | Note |
| responseStatus | Must exist in all responses. |  |
| responseStatus.success | Signed integer. 0 is no errors.  > 0 is warnings.  < 0 is fatal errors and the response is to be discarded. | Should we define some standard errors and warnings?  Yes, example  1-access denied  2-not supported version  3-input data missing  4-input data wrong format |
| responseStatus.statusMsg | Description of the warning or error. This should only be used by humans for debugging and not by the parser. |  |
| customers | A list of requested customers. |  |
| customers.id | Customer id. |  |
| customers.name | The name of the customer within BAS. |  |
| customers.resources | A list of resources this customer has created. |  |
| customers.resources.id | A unique id (GUID/UUID) for the resource. |  |
| customers.resources.name | The name of the resource in BAS.  A rename of a resource in BAS should not alter the id.  ***Optional:*** *if BCS wants to rename its resource/zone name automatically.*  ***Optional:*** *if the BAS has hierarchic architecture and wants to display the full path. It should in that case be separated with a /* |  |

*example:*

{

"responseStatus": {

"success": 0,

"statusMsg": "ok"

},

"customers": [{

"id": "e3941203-37c8-4aaf-a10c-a46100ccb787",

"name": "Customer name",

"resources": [{

"id": "5817c100-d599-4f2e-9c25-07e7a64075a0",

"name": "Location/resource one"

}, {

"id": "fe77c299-980e-49a2-82a5-4f42a4cadf34",

"name": "Location/resource two"

}]

}]

}

**Fetching resource data (string time and dates)**

**REQUEST**

|  |  |  |
| --- | --- | --- |
| Name | Description | Note |
| method | See method: “GetCustomerData” for more info. |  |
| version | See method: “GetCustomerData” for more info. |  |
| appId | See method: “GetCustomerData” for more info. |  |
| dateFormat | Specifies what type of format the date and times must have in this request. Both the request and response must conform to selected format.  “string” = Dates and times is to be represented as string in the format “yyyy-mm-dd hh:MM:ss”  “epoch” = Dates and times is to be represented as Unix epoch times. The number of seconds that have elapsed since January 1, 1970 (midnight UTC/GMT) It must interpreted as UInt64 |  |
| start | Specifies from what date and time (GMT time) the bookings should start.  Every booking that has it’s start and/or end date **on or after** this time.  start shall be formatted according to dateFormat  2000-01-01 00:00:00≤start<9999-12-31 23:59:59  0<(end-start)≤one year |  |
| end | Specifies from what date and time (GMT time) the bookings should end.  Every booking that has it’s start and/or end date **before** this time.  end shall be formatted according to dateFormat  2000-01-01 00:00:00<end≤9999-12-31 23:59:59  0<(end-start)≤one year |  |
| resources | A list of requested resources.  The resources id is requested with method: “GetCustomerData” in beforehand. |  |

*example:*

{

“method”: “ GetResourceData”,

”version”: 1,

“appId”: “4e481c08-c808-4188-b128-20a5a02ebf96”,

”dateFormat”: ”string”

"start": "2015-05-01",

"end": "2016-05-01",

"resources": [

"5817c100-d599-4f2e-9c25-07e7a64075a0",

"fe77c299-980e-49a2-82a5-4f42a4cadf34"

]

}

**RESPONSE**

|  |  |  |
| --- | --- | --- |
| Name | Description | Note |
| responseStatus | See method: “GetCustomerData” for more info. |  |
| responseStatus.success | See method: “GetCustomerData” for more info. |  |
| responseStatus.statusMsg | See method: “GetCustomerData” for more info. |  |
| list | The list of requested bookings.  (  Booked start-time < requested end-time  and Booked end-time > requested start-time  )  There is a risk with the suggested format that bookings are missed.  Example: If there is a long booking (all week 2016-09-26 to 2016-10-02) and you pull “todays bookings” the system will not return any bookings.  JEFF suggest that the original suggestion from Robin is used. |  |
| list.id | Unique id for this specific booking instance.  No duplicates!  This is an “instance id”, not booking id.  If the booking is a repeated booking “I.e every Sunday” every Sunday must produce its own unique id. This is to accommodate the future function to let the BCS report back information on past bookings to the BAS. |  |
| list.start | The start (GMT time) of the booking.  Expressed as selected dateFormat.  list.end>list.start |  |
| list.end | The end (GMT time) of the booking.  Expressed as selected dateFormat.  list.end>list.start |  |
| list.created | The date and time (GMT time) current booking was made in the BAS.  Expressed as selected dateFormat. |  |
| list.signature | A signature/name on the person who made the current booking. |  |
| list.heat | Indicates what type of heat/temperature the booker wants on this particular booking.  In the BAS this should be a dropdown or simular presented to the booker when the booking is made or the the template is defined.  All values smaller than zero is predefined temperatures in BCS by the facility manager.  -2 = Cleaning temperature  -1 = No heat. The temperature should be the same as if there was no booking at this time. The humidity protection must also be active. This value is meant to be used when the resource is booked but no people will be there.  0 = Standard heat. No change. The predefined temperature is to be used.  > 0 = Selected temperature. I.e value of 21 means that the desired temperature for this booking is 21°C |  |
| list.title | The title of this booking.  The title shall be max 50 charecters | A max length should be possible to use since this is a title and not a novel. |
| list.resources | A list of resources that is booked in the same booking. This is for future functions. Like lightings and doors. |  |

*example:*

{

"responseStatus": {

"success": 0,

"statusMsg": "ok"

},

"list": [{

"id": "afff431b-2835-45a1-9c5e-a100d746ea0c",

"start": "2015-05-05 11:30:00",

"end": "2015-05-05 12:00:00",

”created”: ”2015-05-01 11:00:00”,

”signature”: ”Eva Andersson”,

"heat": 0,

"title": "Booking with standard heat temp",

"resources": [

"fe77c299-980e-49a2-82a5-4f42a4cadf34"

]

}, {

"id": "99c42508-0e9c-4eef-af19-d7dbb48f9b27",

"start": "2015-04-30 18:30:00",

"end": "2015-05-02 19:30:00",

”created”: ”2015-04-01 11:00:00”,

”signature”: ”Eva Andersson”,

"heat": -1,

"title": "Booking with no heat",

"resources": [

"5817c100-d599-4f2e-9c25-07e7a64075a0",

"0f01ce5b-0c9b-4534-b1f1-f4891511ecb1"

]

}]

}

**Document version notes**

**Notes 0.0.3:**

**1.** ROWS

After a discussion between ROWS and Jeff a decision has been made to remove timezones.

The goal is to always express dates and times in GMT (Greenwich Mean Time).

It will be up to the receiver of the data to convert the time to appropriate timezone and apply daylight savings if necessary. This is not democratically voted since the number of introduced companies is too low.

**2.** ROWS

More information about each parameter is requested from Jeff. This is now applied.

**Document version notes**

**Notes 0.0.4:**

**1.** Eniac Data

Eniac Data AB added to list.

**Document version notes**

**Notes 0.0.5:**

**1.** ROWS

Added LJ System AB and JEFF Electronics in the list.

**2.** ROWS

Changed ROWS color to Yellow and gave LJ System the orange.

**3.** ROWS

Approved minor clarifying texts from Eniac.

**4.** ROWS

Adopted Eniacs suggestion on the extra information for “The list of requested bookings.”

**Document version notes**

**Notes 0.0.6:**

**1.** ROWS

Added Mildmedia in the list.

**2.** ROWS

Moved Document version notes to the end of the document.

**Notes 0.0.7:**

**1.** JEFF

Added some small changes