Especificación de los servicios de las alarmas

En todos los casos se invocará a un servicio enviando parámetros en formato json y recibiendo la respuesta en el mismo formato.

El parámetro GUID deberá obtenerse desde alguno de los sistemas externos, posiblemente de thalamus.

A continuacián se listan los servicios, con el uso, los parámetros que se enviarán y el formato de la respuesta.

Servicios a invocar a GIS

getAlarms.json

Propósito: Obtener las alarmas de un usuario.

Parámetro:

{"guid": "1"}

Respuesta:

[

{

"description": "Casa",

"id": "1",

"lastChangeAction": "ACTIVAR",

"lastChangeDate": "2013-03-19",

"lastChangeHour": "20:30",

"lastChangeUser": "VALENTINA",

"on": **true**

},

{

"description": "Estudio",

"id": "2",

"lastChangeAction": "ACTIVAR",

"lastChangeDate": "2013-02-01",

"lastChangeHour": "20:30",

"lastChangeUser": "MARCELA",

"on": **false**

}

]

sendPanic.json

Propósito: Disparar la señal de pánico sobre una alarma

Parámetro:

{

"guid": "1",

"alarmId": 2

}

Respuesta:

{

"result": true

}

true si pudo dispararse, false caso contrario

activateAlarm.json

Propósito: Activar una alarma

Parámetro:

{

"guid": "1",

"alarmId": 2,

"password": "5748"

}

Respuesta:

{

"result": true

}

true si pudo activarse, false caso contrario

deactivateAlarm.json

Propósito: Desactivar una alarma

Parámetro:

{

"guid": "1",

"alarmId": 2,

"password": "5748"

}

Respuesta:

{

"result": true

}

true si pudo activarse, false caso contrario

getAlarmLog.json

Propósito: Obtener el log de las últimas x acciones sobre una alarma

Parámetro:

{

"guid": "1",

"alarmId": "1"

}

Respuesta:

[

{

"action": "ACTIVAR",

"date": "2012",

"hour": "20:30",

"user": "MARCELA"

},

{

"action": "DESACTIVAR",

"date": "2012",

"hour": "22:30",

"user": "MARCOS"

}

]

Servicios a invocar no GIS

getAlarmAgendas.json

Propósito: Obtener la agenda de una alarma

Parámetro:

{

"guid": "1",

"agendaId": "1"

}

Respuesta:

[

{

"activateTime": "10:00:00",

"deactivateTime": "17:30:00",

"description": "Dias habiles",

"id": "1",

"type": "BUSINESS\_DAYS",

"customDays": "",

"active": **false**,

"from": "2012-08-01",

"to": "2012-08-31"

},

{

"activateTime": "20:00:00",

"deactivateTime": "05:00:00",

"description": "Findes",

"id": "2",

"type": "WEEKENDS",

"customDays": "",

"active": **true**,

"from": "2012-08-01",

"to": "2015-08-31"

},

{

"activateTime": "20:00:00",

"deactivateTime": "05:00:00",

"description": "Findes",

"id": "3",

"type": "CUSTOM",

"customDays": "Lu,Mi,Vi",

"active": **false**,

"from": "2013-01-01",

"to": "2015-08-31"

}

]

Activate y deactivate time iran en formato HH (00 a 23) :MM (minutos) :SS (segundos).

From y to seran fechas en formato YYYY:MM:DD

type podra contener los valores BUSINESS\_DAYS, WEEKENDS, ALL\_DAYS y CUSTOM.

En caso de ir CUSTOM, se deberan tomar los dias de customDays, con la siguiente nomenclatura de Lunes a Domino: Lu, Ma, Mi, Ju, Vi, Sa, Do

deleteAlarmAgenda.json

Propósito: Realizar el borrado lógico/inactivación de una agenda

Parámetro:

{

"guid": "1",

"agendaId": "2"

}

Respuesta:

{

"result": true

}

true si pudo borrarse, false caso contrario

activateAlarmAgenda.json

Propósito: Reactivar una agenda

Parámetro:

{

"guid": "1",

"agendaId": "2"

}

Respuesta:

{

"result": true

}

true si pudo activarse, false caso contrario

addAlarmAgenda.json

Propósito: agregar una agenda

Parámetro:

{

"activateTime": "20:00:00",

"deactivateTime": "05:00:00",

"description": "Findes",

"guid": "1",

"type": "CUSTOM",

"customDays": "Lu,Mi,Vi",

"active": **false**,

"from": "2013-01-01",

"to": "2015-08-31"

}

Respuesta:

{

"result": true

}

true si pudo agregarse, false caso contrario

saveAlarmAgenda.json

Propósito: Crear una nueva agenda

Parámetro:

{

"activateTime": "20:00:00",

"deactivateTime": "05:00:00",

"description": "Findes",

"guid": "1",

"id": "1",

"type": "CUSTOM",

"customDays": "Lu,Mi,Vi",

"active": **false**,

"from": "2013-01-01",

"to": "2015-08-31"

}

Respuesta:

{

"result": true

}

true si pudo salvarse, false caso contrario

getAlarmAlertConfiguration.json

Propósito: Obtener las configuraciones de alertas por email para una alarma

Parámetro:

{

"guid": "1",

"alarmId": "1"

}

Respuesta:

{

"activateDeactivate": false,

"activateDeactivateByOther": true,

"alarmActivation": true,

"powerSupplyLost": true

}

Aqui queda por ver como manejar el tema de los otros usuarios.

saveAlarmAlertConfiguration.json

Propósito: Salvar las configuraciones de alerts por email para una alarma

Parámetro:

{

"guid": "1",

"activateDeactivate": false,

"activateDeactivateByOther": true,

"alarmActivation": true,

"powerSupplyLost": true

}

Respuesta:

{

"result": true

}

true si pudo salvarse, false caso contrario

Dos temas pendientes de definir son:

1. De donde saldrian y como se usarian en las alertas por email el activar/desactivar por otros usuarios, quizas en la configuracion deberian venir en un atributo el listado de los otros usuarios separados por coma.
2. Manejo de errores puntuales: Se deberan manejar errores funcionales, es decir, podra retornarse un error que indique al usuario que debe revisar? Esto parece no aparecer en mas de un lado, pero quizas podrian codificarse, por ejemplo, si se quiere salvar una alarma con fecha de fin anterior al inicio. Una opcion seria que en la respuesta de error, se liste en un array los codigos de errores, y luego se le presenten al usuario.

Queda pendiente de definir los servicios relativos a luces, pero estos seran muy similares a los definidos.