



Integração do GSAN com o sistema SIGO

Documentação sobre a integração do **GSAN** com o sistema **SIGO**, cujo objetivo é definir a estrutura de troca de informações entre o **GSAN** e o sistema **SIGO**. Aqui é definida a estrutura dos *payloads* que será utilizada nas *request* para o sistema **SIGO**.

Estrutura dos Json de envio

Serão realizadas 2 (duas) requisições à API do sistema **SIGO**:

1.
A primeira, quando for informada a interrupção da rede na execução do serviço para um tipo de serviço no sistema **GSAN**.
2.
A segunda requisição se dá quando o serviço for encerrado em campo e as informações forem atualizadas no **GSAN**.

Requisição para Inserir Ordem Serviço no SIGO

O **Json** será constituído pelos campos abaixo descritos:

1.
id: identificador da requisição enviada. Campo inteiro.
2.
latitude: coordenada geográfica referente ao eixo **x** do endereço informado no serviço. Campo decimal de tamanho **24** com precisão **12**.
3.
longitude: coordenada geográfica referente ao eixo **y** do endereço informado no serviço. Campo decimal de tamanho **24** com precisão **12**.
4.
numeroRA: código do Registro de Atendimento. Campo inteiro.
5.
dataConhecimento: data de conhecimento do serviço a ser executado. Campo *String* contendo a data no formato dd/mm/aaaa hh24:mm.
6.
dataRegistro: data de registro da Ordem de Serviço. Campo *String* contendo a data no formato dd/mm/aaaa hh24:mm.
7.
numeroOrdemServico: código da Ordem de Serviço. Campo inteiro.
8.
codigoTipoServico: código do serviço solicitado. Campo inteiro.
9.
descricaoTipoServico: descrição do serviço solicitado. Campo texto com no máximo 30 (trinta) caracteres.
10.
haverInterrupcaoOperacaoRede: indicador se haverá interrupção ou não da rede. Campo *booleano*.
11.
dataPrevistaExecucao: data prevista para execução do serviço. Campo *String* contendo a data no formato dd/mm/aaaa hh24:mm.
12.
previsaoExecucao: tempo previsto para conclusão do serviço. Campo decimal de tamanho **13** e precisão **2**.
13.
tipoPrevisao: código do tipo de previsão, onde **1 = Dias** e **2 = Horas**. Campo inteiro.

Ficando assim definido a estrutura do Json para inserir a Ordem de Serviço no sistema **SIGO**:

```
{ id: Integer,  
  latitude: decimal,  
  longitude: decimal,  
  numeroRA: Integer,  
    dataConhecimento: String,  
    dataRegistro: String,  
    numeroOrdemServico: Integer,  
  codigoTipoServico: Integer,  
  descricaoTipoServico: String,  
  haveriaInterrupcaoOperacaoRede: Boolean,  
  dataPrevistaExecucao: String,  
  previsaoExecucao: decimal,  
  tipoPrevisao: Integer  
}
```

Exemplo:

```
{id: 1,  
  latitude: 2.12312312,  
  longitude: 2.112312,  
  numeroRA: 1234,  
    dataConhecimento: '12/07/2019 14:34',  
  dataRegistro: '13/07/2019 15:27',  
    numeroOrdemServico: 123443,  
  codigoTipoServico: 99999,  
  descricaoTipoServico: 'SERVICO MANUTENCAO REDE',  
  haveriaInterrupcaoOperacaoRede: true,  
  dataPrevistaExecucao: '15/07/2019 15:00',  
  previsaoExecucao: 1.5,  
  tipoPrevisao: 1  
}
```

Requisição para Encerrar Ordem Serviço no SIGO

O **Json** será constituído pelos campos abaixo descritos:

1.
id: identificador da requisição de inserção enviada anteriormente. Campo inteiro.
2.
numeroOrdemServico: código da Ordem de Serviço. Campo inteiro.
3.
dataSaidaEquipe: data/hora de saída da equipe. Campo *String* contendo a data/hora no formato dd/mm/aaaa h24:mm.
4.
dataInicioExecucao: data/hora de início da execução do serviço. Campo *String* contendo a data/hora no formato dd/mm/aaaa h24:mm.
5.
dataConclusaoExecucao: data/hora de conclusão da execução do serviço. Campo *String* contendo a data/hora no formato dd/mm/aaaa h24:mm.
6.
situacaoOrdemServico: código da Situação da execução da Ordem de Serviço, onde **1 = Total**, **2 = Parcial** e **3 = Cancelada**. Campo inteiro.
7.
equipe: equipe que executou o serviço em campo, o qual será composto por:
 1.
código: código da equipe. Campo inteiro.
 2.
nome: descrição da equipe. Campo *String* com no máximo 30 caracteres.
 3.
componentes: lista de funcionários que compõem a equipe, sendo definidos pelos campos abaixo:
 1.
matricula: matrícula do funcionário. Campo inteiro.
 2.
nome: nome do funcionário. Campo texto com no máximo 45 (quarenta e cinco) caracteres.
8.
quantidadeHorasTrabalhadas: quantidade de horas de execução do serviço. Campo decimal com tamanho **13** e precisão **2**.
9.
insumos: lista de materiais utilizados na execução do serviço, podendo ser uma lista vazia, onde:
 1.
código: código do material. Campo inteiro.
 2.
descrição: descrição do material. Campo texto com no máximo 30 (trinta) caracteres.
 3.
quantidade: quantidade utilizada. Decimal com tamanho **13** e precisão **2**.
10.
codigoTipoMaterialRede: código do tipo de material de rede. Campo inteiro; nulo caso não informado.
11.
descricaoTipoMaterialRede: descrição do tipo de material de rede. Campo texto com no máximo 30 (trinta) caracteres; nulo caso não informado.
12.
codigoDiametroRede: código do diâmetro da rede. Campo inteiro; nulo caso não informado.
13.
descricaoDiametroRede: descrição do diâmetro da rede. Campo texto com no máximo 30 (trinta) caracteres; nulo caso não informado.
14.
profundidadeRede: profundidade da rede. Campo decimal com tamanho **13** e precisão **2**.

Ficando assim definido a estrutura do **Json** para encerrar a Ordem de Serviço no sistema **SIGO**:

```
{id: Integer,  
numeroOrdemServico: Integer,  
dataInicioServico: Date,  
dataConclusaoServico: Date,  
situacaoExecucaoServico: Integer,  
equipe:  
  
{id: Integer,  
nome: String(  
componentes: [{  
matricula: Integer  
nome: String(45)  
}],  
...  
},  
quantidadeHorasTrabalhadas: Integer,  
insumos: [{  
codigo: Integer  
descricao: String(45)  
quantidade: decimal(13,2)  
},  
...  
],  
codigoTipoMaterialRede: Integer,  
descricaoTipoMaterialRede: String,  
codigoDiametroRede: Integer,  
descricaoDiametroRede: String,  
codigoProfundidadeRede: Integer,  
descricaoProfundidadeRede: String  
}
```

Exemplo:

```
{id: 1,  
numeroOrdemServico: 123443,  
dataInicioServico: '15/07/2019 15:22',  
dataConclusaoServico: '15/07/2019 17:22',  
situacaoExecucaoServico: 1,  
equipe:  
  
{  
id: 1,  
nome: 'EQUIPE 01',  
componentes: [{  
matricula: 1  
nome: 'FUNCIONARIO 01'  
}],  
}
```

```
{
matricula: 2
nome: 'FUNCIONARIO 02'
},
quantidadeHorasTrabalhadas: 2.0,
insumos: [{
codigo: 1
descricao: 'MATERIAL 01'
quantidade: 1.5
},
],
codigoTipoMaterialRede: 1,
descricaoTipoMaterialRede: 'MATERIAL REDE 01',
codigoDiametroRede: null,
descricaoDiametroRede: null,
codigoProfundidadeRede: 1,
descricaoProfundidadeRede: 'PROFUNDIDADE 01'
}
```

Requisição de Retorno do SIGO

Para toda requisição que for enviada ao **SIGO**, semelhante requisição deve retornar um **Json** informando que a *request* foi recebida com sucesso. Este retorno deve conter os campos abaixo descritos:

1.
id: identificador da requisição enviada para processamento no **SIGO**.
2.
status: situação da requisição, onde 1 = recebido com sucesso.

A estrutura do **Json** do retorno das requisições, fica assim definida:

```
{id: Integer
status: Integer
}
```

Exemplo:

```
{id: 1
status: 1
}
```

Clique **AQUI** para acessar outros serviços com os detalhes técnicos das integrações.

From:

<https://www.gsan.com.br/> - **Base de Conhecimento de Gestão Comercial de Saneamento**

Permanent link:

<https://www.gsan.com.br/doku.php?id=integracoes:api-sigo&rev=1568356951>

Last update: **13/09/2019 06:42**

