
Manual de sincronismo de dados da automação

fev. 25, 2022

1	Passos para receber os arquivos JSON	1
1.1	Servidor ssh/rsync	1
1.2	Liberar acesso do IP	1
1.3	Enviar configurações	1
1.4	Adicionar chave pública	2
2	Processamento dos arquivos	3
2.1	Exemplo simples de processamento	3
2.2	Estrutura do JSON	3
2.3	Documentação dos campos	5
3	Índices	9

Passos para receber os arquivos JSON

- Servidor ssh/rsync;
- Liberar acesso do IP;
- Enviar configurações;
- Adicionar chave pública;

1.1 Servidor ssh/rsync

É necessário um servidor com o ssh habilitado e o rsync instalado.

Geralmente o rsync já vem instalado nas distribuições do Linux, mas se for necessário instalar as instruções verifique a forma de instalar no seu sistema. O manual online do rsync se encontra em <https://linux.die.net/man/1/rsync>

1.2 Liberar acesso do IP

O IP que iremos utilizar como saída das informações é 18.229.104.253.

Atenção: O servidor não está conectado na INFOVIA. Não é possível a conexão direta via INFOVIA.

1.3 Enviar configurações

Ao mesmo tempo, precisamos saber qual é o endereço do servidor de vocês e o número da porta que será aberta para a conexão SSH. Além disso, a pasta que iremos gravar o arquivo e o usuário que vai receber. O usuário precisa de permissão de escrita na pasta.

1.4 Adicionar chave pública

Ao enviar as configurações iremos enviar para você a chave pública para acesso automatizado. Ver o passo 3 do guia como instalar <https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys-2>

Importante: Por favor verificar a permissão na pasta .ssh e no arquivo authorized_keys ao adicionar a chave pública. A permissão deve ser a seguinte: 755 no .ssh 600 no authorized_keys O usuário do ssh deve ser o usuário da pasta .ssh e o usuário do ssh deve estar no grupo.

Importante: Não envie a senha via email. Não há necessidade de senha uma vez que a chave pública foi instalada. Caso deseje nos informar a senha faça por um meio seguro.

Atenção: O arquivo vai ser sempre adicionado da diferença. Caso deseje ter um histórico é necessário copiar o arquivo diariamente para outro local.

Processamento dos arquivos

Você pode processar os arquivos da forma que desejar.

Uma forma é utilizar o inotify API ou inotify-tools para processar quando um arquivo for escrito.

2.1 Exemplo simples de processamento

Listagem 1: Espera o evento de modificação e processa o arquivo. Leitura não dispara o processo.

```
#!/bin/sh
while inotifywait -e modify /meu/arquivo/json; do
    if tail -n1 /meu/arquivo/json | grep id=33; then
        /meu/processamento/processa.py /meu/arquivo/json
    fi
done
```

Veja o manual do inotify-tools: <https://linux.die.net/man/1/inotifywait>

Wiki do inotify-tools: <https://github.com/inotify-tools/inotify-tools/wiki>

2.2 Estrutura do JSON

```
{
  "id": 0,
  "form_versao": "<VALOR PREENCHIDO>",
  "IDE_FINALIZADO": "<VALOR PREENCHIDO>",
  "DES_TITULO": "<VALOR PREENCHIDO>",
  "metadados": [
    {
      "COD_PROCESSO_F": 0,
```

(continues on next page)

```

"COD_ETAPA_F": 0,
"COD_CICLO_F": 0,
"CPF": "<VALOR PREENCHIDO>",
"PROTOCOLO": "<VALOR PREENCHIDO>",
"<DADOS DO FORMULARIO>": "....",
  "TITULO_ETAPA": "<VALOR PREENCHIDO>",
  "PROCESSOS_ETAPAS": {
    "COD_CICLO": 0,
    "COD_ETAPA": 0,
    "APROVACAO_MOB": "<VALOR PREENCHIDO>",
    "COD_USUARIO_ETAPA": 0,
    "DAT_FINALIZACAO": "<VALOR PREENCHIDO>",
    "DAT_GRAVACAO": "<VALOR PREENCHIDO>",
    "IDE_STATUS": "<VALOR PREENCHIDO>",
    "IDE_STATUS_ANT": "<VALOR PREENCHIDO>",
    "IDE_TEMPORARIO": "<VALOR PREENCHIDO>",
    "IDE_VISUALIZADO": "<VALOR PREENCHIDO>",
    "NOM_ARQ_ASSINADO": "<VALOR PREENCHIDO>",
    "uuid": "<VALOR PREENCHIDO>",
    "VLR_ATRASO1": 0,
    "VLR_CAMPOS_COMPLEM": "<VALOR PREENCHIDO>",
    "VLR_QTDE_ALERTA": 0,
    "VLR_QTDE_ATRASO": 0,
    "VLR_TEMP_ATR_GEST": 0,
    "VLR_TEMP_HIBERNA": 0,
    "VLR_TEMP_LIM_GEST": 0,
    "VLR_TEMPO_ALERTA": 0,
    "VLR_TEMPO_ATRASO": 0,
    "VLR_TEMPO_CONSUMIDO": 0,
    "VLR_TEMPO_CONSUMIDO_CORRIDO": 0,
    "VLR_TEMPO_ESCALONA": 0,
    "COD_PROCESSO": 0,
    "DATA_PRINT": 0
  }
},
"NOME_DA_GRID": [
  {
    "identificador": 0,
    "COD_PROCESSO": 0,
    "COD_ETAPA": 0,
    "COD_CICLO": 0,
    "INDICE": 0,
    "<DADOS DA GRID DO FORMULARIO>": "....",
    "TITULO_ETAPA": "<VALOR PREENCHIDO>"
  }
],
"NOME_DE_OUTRA_GRID": [
  {
    "identificador": 0,
    "COD_PROCESSO": 0,
    "COD_ETAPA": 0,
    "COD_CICLO": 0,
    "INDICE": 0,
    "<DADOS DA GRID DO FORMULARIO>": "....",
    "TITULO_ETAPA": "<VALOR PREENCHIDO>"
  }
]

```

(continues on next page)

(continuação da página anterior)

```
}  
  ]  
}
```

Nota: Cada JSON tem dados específicos pois é referente a um serviço único. Verifique com o gestor do serviço quais são os campos dos formulários das etapas/ciclos.

Os nomes dos campos estão separados pelo caractere “|”.

No formato: <NOME_DO_CAMPO_NO_FORMULARIO>|<LABEL_DO_CAMPO_NO_FORMULARIO_SE_EXISTIR>
ou só o <NOME_DO_CAMPO_DO_FORMULARIO>

2.3 Documentação dos campos

2.3.1 Raiz do JSON

Composto de um array com N solicitações do serviço público.

Campos

id - int(11) PK Código do processo

form-versao - varchar(20) Formulário e Versão (cod_form e cod-versao)

ide-finalizado - varchar(1)

Flag do status do processo;

- A -> Em andamento;
- P -> Aprovado;
- C -> Cancelado;
- R -> Rejeitado.

des-titulo - varchar(50) Nome do modelo BPM

2.3.2 Metadados

Composto de um array com N etapas da solicitação.

Campos

cod-processo-f - int(11) PK Código do processo

cod-etapa-f - smallint(6) PK Código da etapa

cod-ciclo-f - smallint(6) PK Código do ciclo

cpf - varchar(13) Cpf do cidadão

protocolo - varchar(19) Protocolo

<**DADOS DO FORMULARIO**> Dados do formulário conforme documentação da automação do serviço homologado pelo órgão. Os formulários podem ser acessados na interface de atendimento disponibilizada ao dono do serviço.

titulo_etapa - varchar(60) Nome da etapa

processos_etapas Conjunto de informações para cada etapa da solicitação e metadados.

Processos_etapas

cod_ciclo - smallint(6) PK Código do ciclo

cod_etapa - smallint(6) PK Código da etapa

aprovacao_mob - varchar(255) Aprovação Mob

cod_usuario_etapa - smallint(6) PK Código do usuário da etapa. O usuário aqui é o servidor público resolvidor dessa etapa e **não é o cidadão**.

dat_finalizacao - datetime Data de conclusão da etapa (envio do formulário)

dat_gravacao - datetime Data de início da etapa (data de conclusão da etapa anterior ou data de abertura)

ide_status - varchar(255)

Flag do status da etapa.

- A -> Em andamento;
- D -> Escalonado;
- P -> Aprovado;
- R -> Rejeitado;
- C -> Cancelado;
- H -> Hibernado;
- L -> Em paralelismo.

ide_status_ant - varchar(255)

Flag do status anterior da etapa.

- A -> Em andamento;
- D -> Escalonado;
- P -> Aprovado;
- R -> Rejeitado;
- C -> Cancelado;
- H -> Hibernado;
- L -> Em paralelismo.

ide_temporario - varchar(255) Status temporário

ide_visualizado - varchar(255) Status visualizado

nom_arq_assinado - varchar(255) Nome do arquivo assinado

uuid - varchar(255) Unique ID da etapa

vlr_atraso1 - bigint(20) Valor de atraso

vlr_campos_complem - **varchar(255)** Valor campos complementados
vlr_qtde_alerta - **int(11)** Valor quantidade de alerta
vlr_qtde_atraso - **int(11)** Valor quantidade de atraso
vlr_qtde_atraso_gest - **int(11)** Valor quantidade de atraso gestor
vlr_temp_atr_gest - **bigint(20)** Valor tempo atraso gestor
vlr_temp_hiberna - **bigint(20)** Valor tempo hibernação
vlr_temp_lim_gest - **bigint(20)** Valor tempo limite gestor
vlr_tempo_alerta - **bigint(20)** Valor tempo alerta
vlr_tempo_atraso - **bigint(20)** Valor tempo atraso
vlr_tempo_consumido - **bigint(20)** Valor tempo consumido
vlr_tempo_consumido_corrido - **bigint(20)** Valor tempo consumido corrido
vlr_tempo_escalona - **bigint(20)** Valor tempo escalonado
cod_processo - **int(11)** **PK** Código do processo
data_print - **longtext** Data print
des_login - **varchar(50)** Login do usuário responsável pela etapa
des_email - **varchar(255)** Email do usuário responsável pela etapa
nom_usuario - **varchar(50)** Nome do usuário responsável pela etapa

2.3.3 NOME DA GRID

Composto de um array de N registros da grid para cada etapa da solicitação e metadados.

Campos

identificador - **int(11)** Identificador
cod_processo - **int(11)** **PK** Código do processo
cod_etapa - **smallint(6)** **PK** Código da etapa
cod_ciclo - **smallint(6)** **PK** Código do ciclo
indice - **smallint(6)** Índice do elemento
<**DADOS DA GRID DO FORMULARIO**> Dados da grid do formulário conforme documentação da automação do serviço homologado pelo órgão. As grids podem ser acessadas na interface de atendimento disponibilizada ao dono do serviço.
titulo_etapa - **varchar(60)** Nome da etapa

Nota: Os campos **cod_processo**, **cod_etapa** e **cod_ciclo** são chaves compostas para identificar os registros da grid para cada etapa. O **indice** identifica a ordem do elemento na grid.

Atenção: Cada grid possui o seu próprio nome. Verifique com o dono do serviço quais grids existem na etapa.

CAPÍTULO 3

Índices

- genindex