

Gestor - Integrações

Integração WSTT (SOAP) ou API (REST)



20 anos

ESTRUTURA - RESUMO



Intuito da API do portal Gestor é realizar uma ligação de forma simples a outros sistemas, independente do dispositivo e linguagem de programação, isso se deve ao uso do padrão Webservice (XML).

Possuímos 2 tipos de protocolos:

SOAP: É fornecido dados em sequencia onde integração solicita ao Gestor e consome uma fila de eventos.

REST: De formato diferente é disponibilizado através de “Endpoints”(Consultas personalizadas), porém ao solicitar evento a consulta solicita parâmetros (Ex.: Placa, Data, Situação) e recebe dados.

Observações:

- Integração pelo portal Gestor possibilita consulta direta ao servidor Omnilink, liberando a necessidade de um servidor (Central) no ambiente do cliente.
- Sistema está em constante atualização, portanto a integração irá atender a diversos produtos Omnilink e poderá ser personalizado a necessidade dos clientes ou parceiros.

INTEGRAÇÕES DO PORTAL GESTOR (<http://www.omnilink.com.br>)

Para solicitar seu acesso
favor entrar em contato
com a central de
atendimento Omnilink

API Gestor (Tipo: REST)

{ REST }

Endereço: <https://api.showtecnologia.com/>

Documentação: <https://api.showtecnologia.com/documentacao/frotas>

Utilização: Consultas por Endpoints, se necessário pode solicitar criações customizadas.

WSTT (Tipo: SOAP)



Endereço: <https://wstt.omnilink.com.br/iasws/iasws.asmx?wsdl>

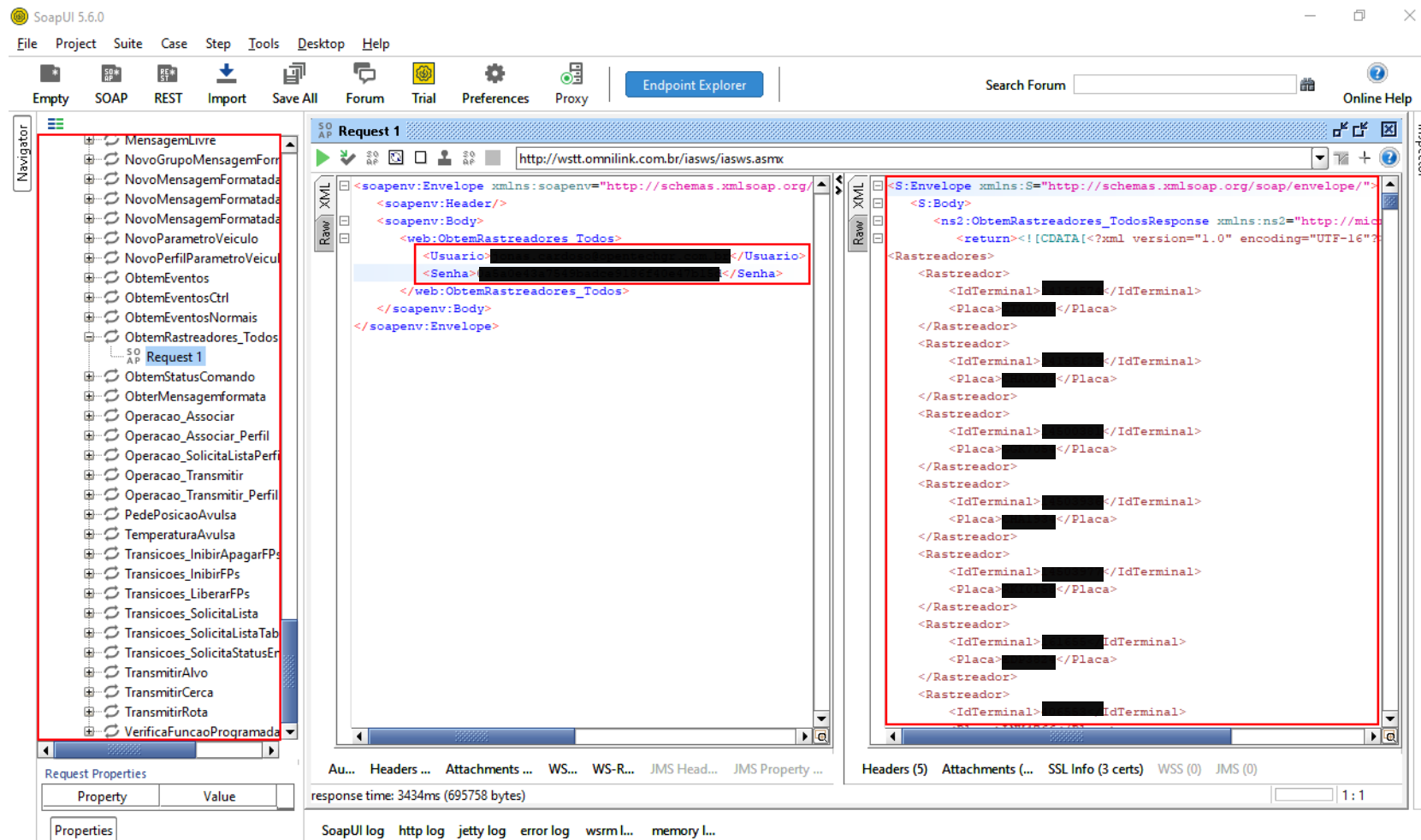
Documentação: http://app.omnilink.com.br/Download/File/5_Manuais_Cliente/Gestor_Manual_de_Integracao_WSTT.pdf

Utilização: Eventos em XML em fila para consumo.

(Mesmos métodos da integração utilizada no IASWS)

Exemplo - WSTT

Utilizando aplicação “SOAP UI” foi simulado uma consulta no método “Rastreadores_Todos” onde mesmo trouxe os XML dos rastreadores com numero de série e placa.

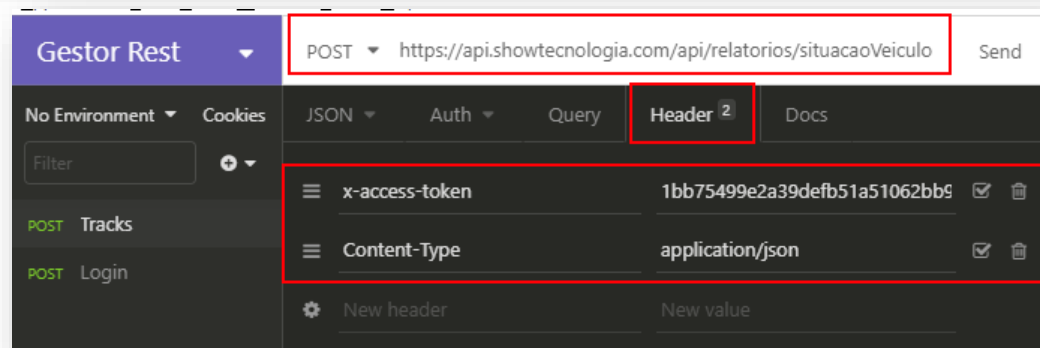
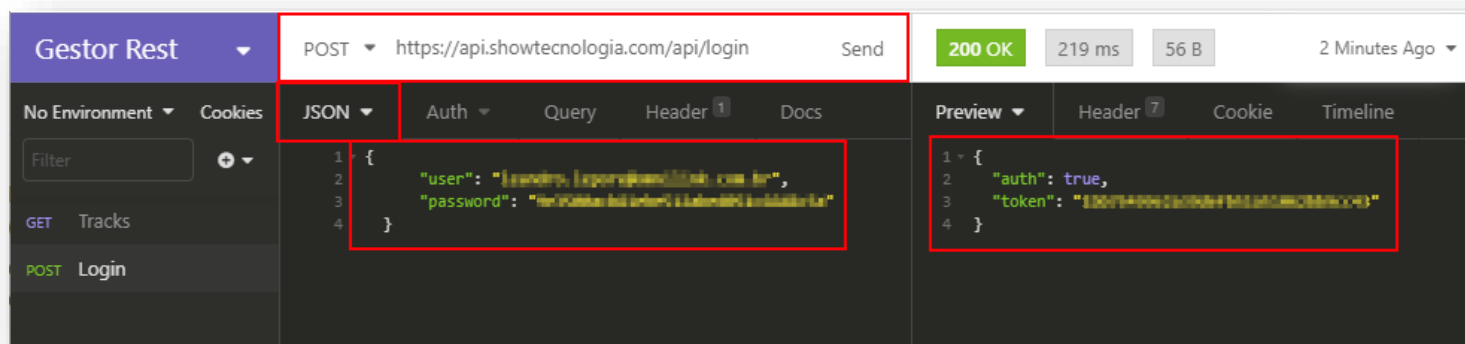


Integração WSTT (SOAP) ou API (REST)

Exemplo – Rest (Acessos)

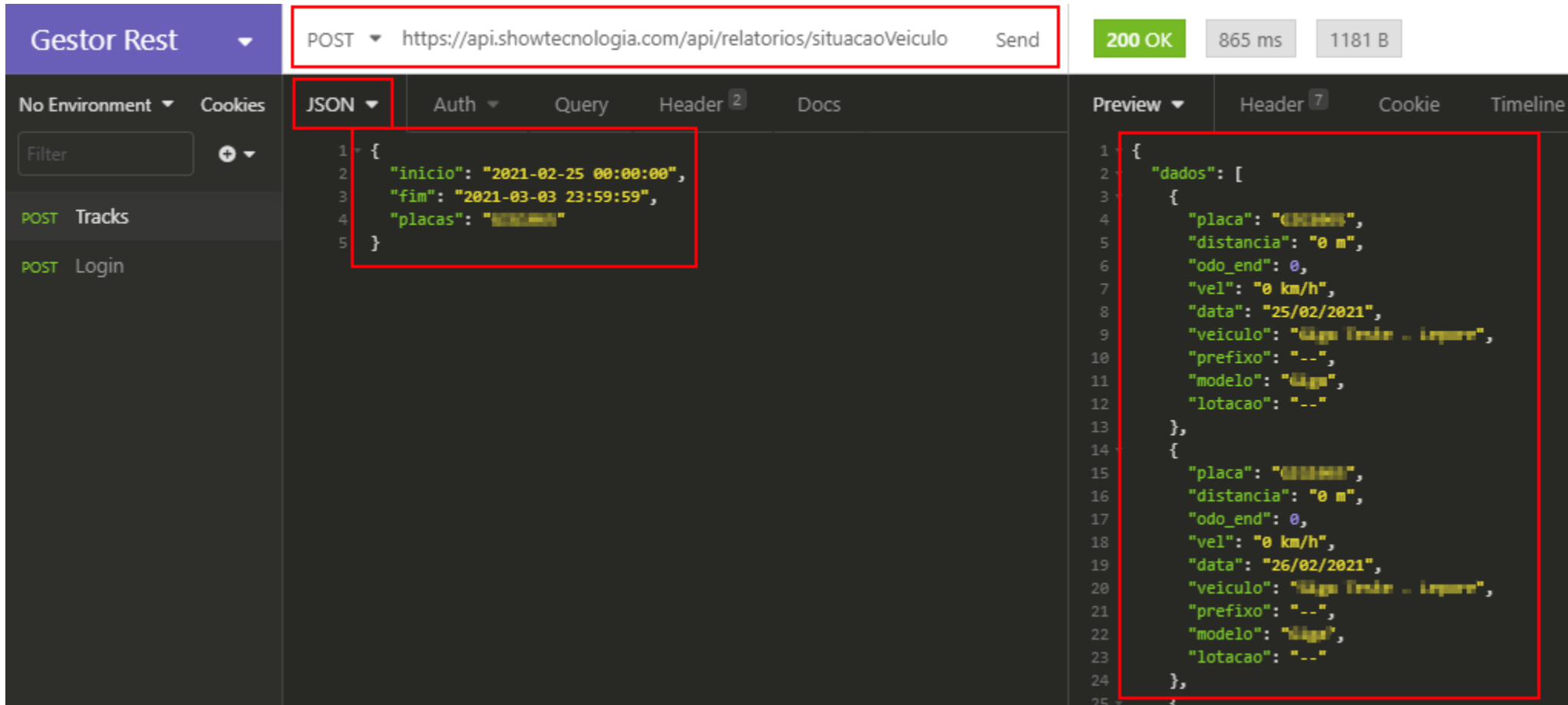
Utilizando aplicação “Insomnia” foi simulado uma consulta no endpoint “/api/login” onde mesmo trouxe os dados da conta de usuários que passamos nos parâmetros “Email” e “Senha” (Senha do portal convertida para MD5).

É fornecido acesso do Token do usuário válido para uso em 24 horas.



Exemplo – Rest (Consulta)

Utilizando aplicação “Insomnia” foi simulado uma consulta no endpoint “api/relatórios/situacaoVeiculo” onde bastou informar campos como informado na pagina de documentação e foi retornado dados em formato JSON.



The screenshot displays the Insomnia REST client interface. At the top, the method is set to POST and the endpoint is `https://api.showtecnologia.com/api/relatorios/situacaoVeiculo`. The status bar indicates a successful response with status 200 OK, a response time of 865 ms, and a body size of 1181 B.

The request body is defined in the JSON tab as follows:

```
1 {
2   "inicio": "2021-02-25 00:00:00",
3   "fim": "2021-03-03 23:59:59",
4   "placas": "XXXX-XX"
5 }
```

The response body, shown in the Preview tab, is a JSON array containing two vehicle status records:

```
1 {
2   "dados": [
3     {
4       "placa": "XXXX-XX",
5       "distancia": "0 m",
6       "odo_end": 0,
7       "vel": "0 km/h",
8       "data": "25/02/2021",
9       "veiculo": "Alga Truck - Igarapê",
10      "prefixo": "--",
11      "modelo": "Alga",
12      "lotacao": "--"
13    },
14    {
15      "placa": "XXXX-XX",
16      "distancia": "0 m",
17      "odo_end": 0,
18      "vel": "0 km/h",
19      "data": "26/02/2021",
20      "veiculo": "Alga Truck - Igarapê",
21      "prefixo": "--",
22      "modelo": "Alga",
23      "lotacao": "--"
24    }
25  ]
26 }
```

Exemplo – Rest (Documentação)

Para acessar pagina de documentação entrar com seu email e senha do Portal Gestor, nele apresenta detalhes para conectar ao API e cada endpoint possui descrição e modelo dos parâmetros para solicitar consulta e informações que sistema retorna.

Documentação API Node

Faça o login na API utilizando um usuário e senha cadastrados no Gestor

Usuário:

Senha:

Entrar

API Frotas

Esta API é usada para a requisição dos registros necessários aos diferentes relatórios que podem ser gerados no Gestor.

Requisitos Para Execução Local

```
node -v
node -js 12.13.1
Caso tenha o Chocolatey instalado, execute chst node --version 12.13.1
yarn (preferencialmente)
chst yarn
memoashed
chst memoached
```

Execução Local da API

Antes de executar a API você deve inicializar o projeto com **yarn install** em sua raiz (".../api-1.0/"). Para executar a API localmente você precisa **manter o memoashed em execução**, após garantir isso, inicie execução a API com **yarn dev**

Visão Geral

Para utilizar esta API você primeiramente precisa acessar o endpoint de **login** para gerar um token de acesso, que deverá ser enviado no header de todas as requisições à API com **key = x-access-token** e **value** igual ao token retornado no login. O token gerado no login é **válido por 24 horas**.

Autenticação

Não é necessária uma autenticação para enviar requisições a esta API, o que você precisa é enviar o token de acesso no header das requisições como mencionado em **Visão Geral**

Códigos de Erro

500 - **Internal Server Error**: Está sendo usado genericamente para erros em geral.
400 - **Bad Request**: Dados da requisição inválidos.
401 - **Unauthorized**: Token inválido, erro no login.

Authorize 

Login

POST /api/login Funcionalidade utilizada para fazer login na api, gerando o token de autorização necessário em requisições do sistema. O password deve ser a senha criptografada em MD5.

Relatórios

POST /api/relatorios/desempenhoExpediente Funcionalidade que retorna dados de desempenho que ocorreram no período enviado.

POST /api/relatorios/deslocamento Funcionalidade que retorna o número total de km rodados, posições, tempo rodando e tempo ocioso para cada chave enviada.

POST /api/relatorios/detalhamentoMotorista Funcionalidade que recebe período e retorna detalhes do motorista.



Mais integrado, conectado e seguro.

20 anos