



Projeto: API de Aprovação de Horas Extras via SGOMobile

Nome do Cliente: SEBRAE NA

Data: 02 de julho de 2019

Sumário

Sumário.....	1
1.1 Objetivo Desse Documento	2
1.2 API para autenticação do usuário no SGOLite	2
1.3 API para consultar Horas Extras	4
1.3.1 JSON padrão retornado pela API	4
1.3.2 Header	4
1.3.3 Método <i>GetSubordinadosHorasExtrasPendentes</i>	5
1.3.3 Método <i>GetHorasExtrasPendentes</i>	6
1.4 API para Aprovar Hora Extra	8
1.4.1 JSON padrão retornado pela API	8
1.4.2 Header	9
1.4.3 Método <i>AprovarHorasExtras</i>	10
1.4.4 Método <i>AprovarHorasExtrasSubordinados</i>	12
1.5 API para consultar Justificativas	14
1.5.1 JSON padrão retornado pela API	14
1.5.2 Header	14
1.5.3 Método <i>GetJustificativasHoraExtra</i>	15
1.6 Histórico de Atualização do Documento	17
1.6.1 Histórico de Versões	17



1.1 Objetivo Desse Documento

Prezados Senhores,

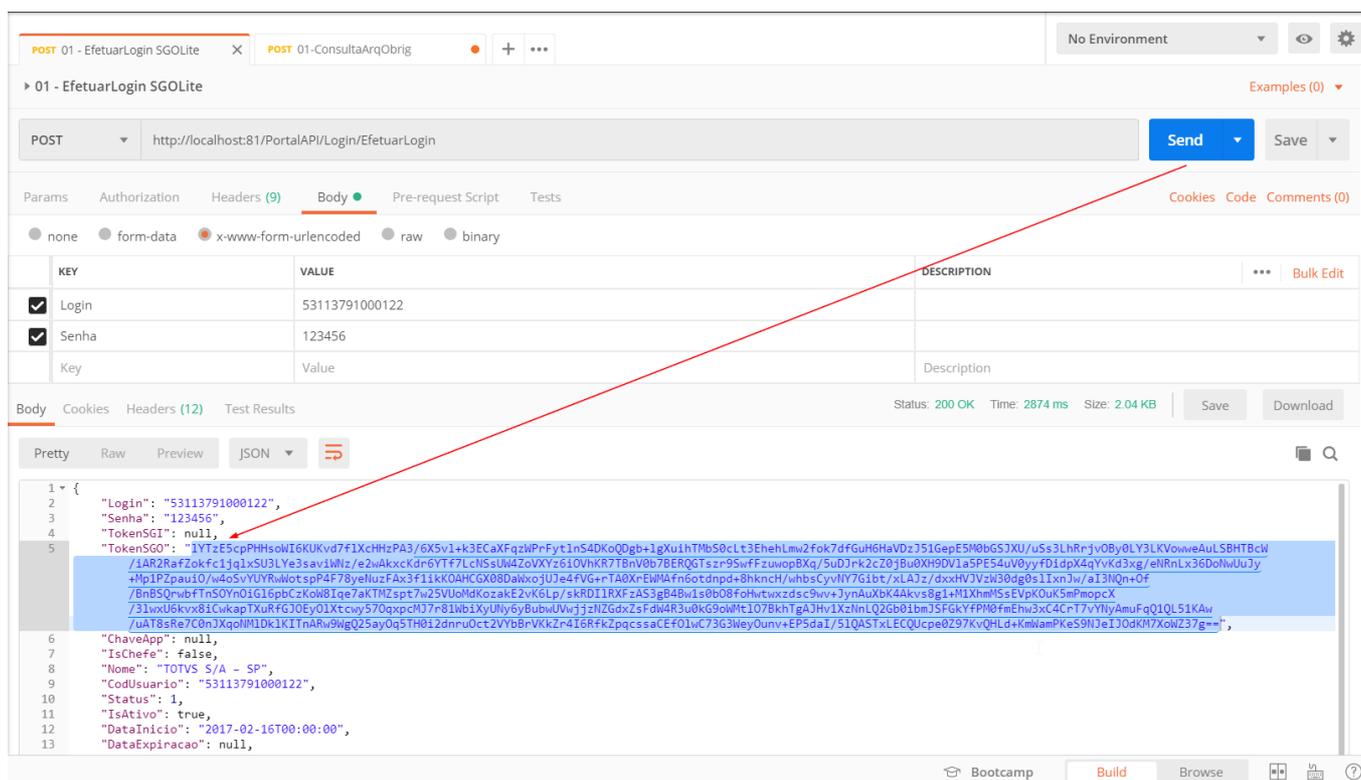
O documento a seguir tem por objetivo orientar o usuário sobre o consumo dos webmetodos da API de Aprovação de Horas Extras do Fornecedor via SGOLite

1.2 API para autenticação do usuário no SGOLite

Antes de executar qualquer API descrita neste documento, deve-se antes realizar a autenticação do usuário na API do SGOLite onde será gerado um toke que deverá ser enviado no cabeçalho das demais requisições.

- **Url:** <http://www.homolog.sgolite.sebrae.com.br/PortalAPI>
- **Controller:** Login
- **Action:** EfetuarLogin

Após efetuar o login com sucesso, a API retornará todas as informações do usuário logado. Nete caso deve-se capturar o valor retornado na propriedade "TokenSGO" para que seja enviado nas próximas requisições da API de Notas Fiscais.



The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:81/PortalAPI/Login/EfetuarLogin
- Params:** none
- Authorization:** x-www-form-urlencoded
- Body:**

KEY	VALUE	DESCRIPTION
Login	53113791000122	
Senha	123456	
Key	Value	Description
- Status:** 200 OK
- Time:** 2874 ms
- Size:** 2.04 KB
- Response Body (JSON):**

```

1 {
2   "Login": "53113791000122",
3   "Senha": "123456",
4   "TokenSGI": null,
5   "TokenSGO": "1YtzE5cpPHHsoI6KUKvd7f1XcHHzPA3/6X5v1+k3ECaXfQzWPrFyt1nS4DKoQDgb+lgXuihTmbS0clt3EehLmw2fok7dGuH6HaVDz351GepE5M0bG5JXU/uSs3LhRrjv0By0LY3LKVowweAulSBHTBckl
6   "ChaveApp": null,
7   "IsChefe": false,
8   "Nome": "TOTVS S/A - SP",
9   "CodUsuario": "53113791000122",
10  "Status": 1,
11  "IsAtivo": true,
12  "DataInicio": "2017-02-16T00:00:00",
13  "DataExpiracao": null,

```



Abaixo seguem dois exemplos de implementação (C# e JQuery Ajax)

GENERATE CODE SNIPPETS



C# (RestSharp) ▾

Copy to Clipboard

```
1 var client = new RestClient("http://localhost:81/PortalAPI/Login/EfetuarLogin");
2 var request = new RestRequest(Method.POST);
3 request.AddHeader("cache-control", "no-cache");
4 request.AddHeader("Connection", "keep-alive");
5 request.AddHeader("content-length", "33");
6 request.AddHeader("accept-encoding", "gzip, deflate");
7 request.AddHeader("Host", "localhost:81");
8 request.AddHeader("Postman-Token", "27a1a56d-5516-4d1c-9d86-2f5247214fde,29420b89-d113-47a6
  -af79-218fe3fbdb13");
9 request.AddHeader("Cache-Control", "no-cache");
10 request.AddHeader("Accept", "*/*");
11 request.AddHeader("User-Agent", "PostmanRuntime/7.11.0");
12 request.AddHeader("Content-Type", "application/x-www-form-urlencoded");
13 request.AddParameter("undefined", "Login=53113791000122&Senha=123456", ParameterType
  .RequestBody);
14 IRestResponse response = client.Execute(request);
```

GENERATE CODE SNIPPETS



JavaScript JQuery AJAX ▾

Copy to Clipboard

```
1 var settings = {
2   "async": true,
3   "crossDomain": true,
4   "url": "http://localhost:81/PortalAPI/Login/EfetuarLogin",
5   "method": "POST",
6   "headers": {
7     "Content-Type": "application/x-www-form-urlencoded",
8     "User-Agent": "PostmanRuntime/7.11.0",
9     "Accept": "*/*",
10    "Cache-Control": "no-cache",
11    "Postman-Token": "27a1a56d-5516-4d1c-9d86-2f5247214fde,c791b5af-8048-4d74-b89c
    -31d069a9eddf",
12    "Host": "localhost:81",
13    "accept-encoding": "gzip, deflate",
14    "content-length": "33",
15    "Connection": "keep-alive",
16    "cache-control": "no-cache"
17  },
18  "data": {
19    "Login": "53113791000122",
20    "Senha": "123456"
21  }
22 }
23
24 $.ajax(settings).done(function (response) {
25   console.log(response);
26 });
```



1.3 API para consultar Horas Extras

Esta API deverá ser utilizada pelos fornecedores para realizar consultas de horas extras pendentes.

1.3.1 JSON padrão retornado pela API

Esta API retornará um JSON padronizado com a seguinte estrutura:

```
{
  "Success": false,
  "Message": "",
  "Data": Objeto enviado
}
```

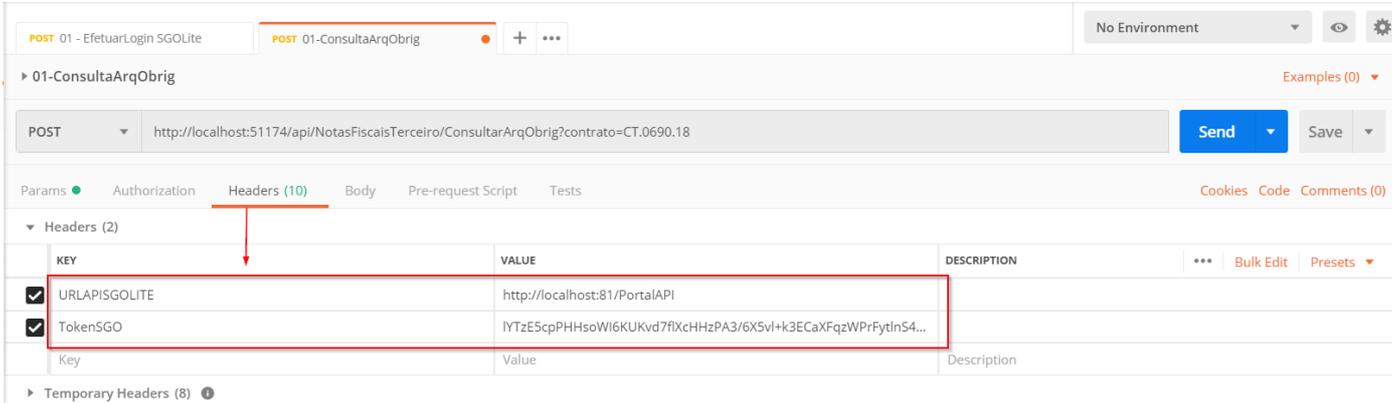
Os campos retornados pelo JSON são descritos na tabela abaixo.

Campo	Tipo de Retorno	Descrição
Success	True/false	Indica se a requisição foi realizada com sucesso.
Message	String	Mensagem de sucesso ou falha apresentada durante a requisição.
Data	Objeto de retorno em JSON	Dados retornados pela requisição em formato JSON

1.3.2 Header

Para consumir a API de consulta por horas extras, deve ser informado no header os campos:

- **URLAPISGOLITE:** Url do Portal Api SGO Lite para autenticação. No caso do ambiente de homologação será: <http://www.homolog.sgolite.sebrae.com.br/PortalAPI/>
- **TokenSGO:** Token referente ao usuário autenticado, para obter esse Token é necessário consumir a API do PortalAPI SGOLite conforme explicado no item 1.2 deste documento



The screenshot shows a REST client interface with the following details:

- Request Method: POST
- URL: `http://localhost:51174/api/NotasFiscaisTerceiro/ConsultarArqObrig?contrato=CT.0690.18`
- Headers Tab: Headers (10)
- Header List:

KEY	VALUE	DESCRIPTION
URLAPISGOLITE	http://localhost:81/PortalAPI	
TokenSGO	lYtzE5cpPHHsoWi6KUKvd7fIXcHHzPA3/6X5vl+k3ECaXFqzWPrFytlN54...	

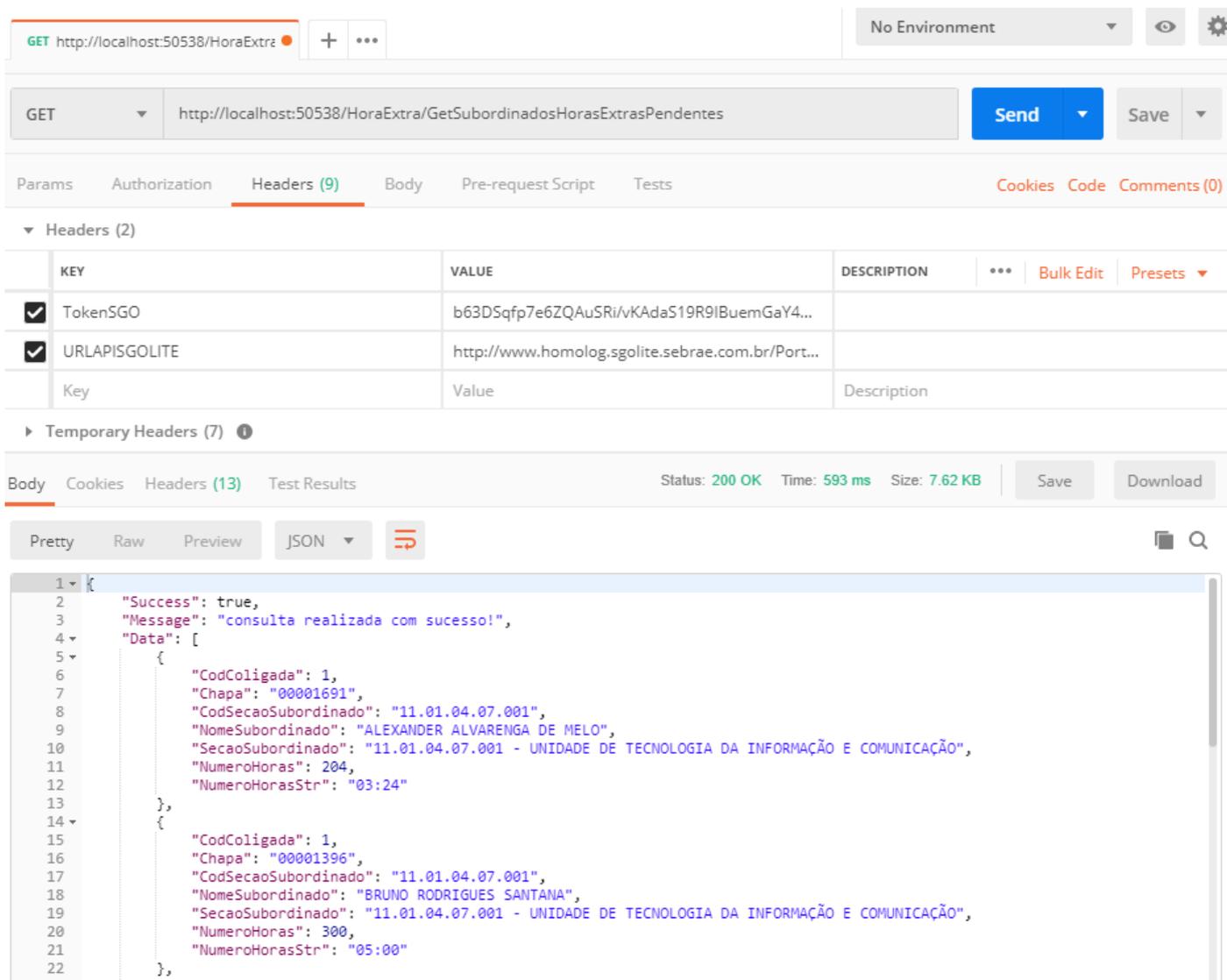


1.3.3 Método GetSubordinadosHorasExtrasPendentes

Método:	GetSubordinadosHorasExtrasPendentes		
Funcionalidade:	Esse método será utilizado para consultar os subordinados com hora extra pendente.		
Verbo HTTP:	GET		
Parâmetros de Entrada (JSON):			
Campo	Tipo de Entrada	Descrição	Obrigatório
		Não se aplica.	

Regras de Negócio da API:

RN1) Ao realizar a requisição, será retornada a lista de subordinados que possuem hora extra pendente de aprovação.



The screenshot shows a REST client interface with the following details:

- Request:** GET `http://localhost:50538/HoraExtra/GetSubordinadosHorasExtrasPendentes`
- Headers (9):**
 - TokenSGO: `b63D5qfp7e6ZQAuSRi/vKAdaS19R9IBuemGaY4...`
 - URLAPISGOLITE: `http://www.homolog.sgolite.sebrae.com.br/Port...`
- Status:** 200 OK, Time: 593 ms, Size: 7.62 KB
- Response (JSON):**

```

{
  "Success": true,
  "Message": "consulta realizada com sucesso!",
  "Data": [
    {
      "CodColigada": 1,
      "Chapa": "00001691",
      "CodSecaoSubordinado": "11.01.04.07.001",
      "NomeSubordinado": "ALEXANDER ALVARENGA DE MELO",
      "SecaoSubordinado": "11.01.04.07.001 - UNIDADE DE TECNOLOGIA DA INFORMACÃO E COMUNICACÃO",
      "NumeroHoras": 204,
      "NumeroHorasStr": "03:24"
    },
    {
      "CodColigada": 1,
      "Chapa": "00001396",
      "CodSecaoSubordinado": "11.01.04.07.001",
      "NomeSubordinado": "BRUNO RODRIGUES SANTANA",
      "SecaoSubordinado": "11.01.04.07.001 - UNIDADE DE TECNOLOGIA DA INFORMACÃO E COMUNICACÃO",
      "NumeroHoras": 300,
      "NumeroHorasStr": "05:00"
    }
  ]
}

```



Abaixo segue exemplo de implementação (C#).

GENERATE CODE SNIPPETS
✕

C# (RestSharp) ▾
Copy to Clipboard

```

1  var client = new RestClient("http://localhost:50538/HoraExtra
   /GetSubordinadosHorasExtrasPendentes");
2  var request = new RestRequest(Method.GET);
3  request.AddHeader("Postman-Token", "25027b55-fafc-4cc1-a410-b17a4c064157");
4  request.AddHeader("cache-control", "no-cache");
5  request.AddHeader("Content-Type", "application/json");
6  request.AddHeader("URLAPISGOLITE", "http://www.homolog.sgolite.sebrae.com.br/PortalAPI");
7  request.AddHeader("TokenSGO", "b63DSqfp7e6ZQAuSRi/vKAdaS19R9IBuemGaY404ukDxcg+Kc5FJ1
   /Yw1wagfW41F1qkJKoXX/SV74KhUyXnIVFjHdfbrBjaTzrJ4dzLq9z/hhwErMXYPftJ+vX1kM4m80GQ9B3
   /2xm29YLCxgA+YsCmba7S07M7Cao4yHw4H0zz9GfVrxhR4EUsRDZBYJesu57a6hGDW7iy16kM1psdw
   +JAgcV6bmf5SnbqIA6Ug2sFDZxScH46ieDzvRP7d9M95jZodi8UpwDD6Twh1wGhDqkZC9gu0Ld5uzk1Wlogki3wy
   a5VIUZWCrblFndAi/9kbCe/LwPOak/frxuxsKo8rLaxzInu7cbshVXiWnY3FnrQ
   +5TYIH8CWRk1egL1SjnzG3skvMsV61ZAFmpftEWFewXBQ4FF0vAI1kwwYFZJi
   +OE7qVZRssZ7a7CfK20188TMBPwy9ZveUoc9tMSXxJR/SFy+6Rq+Qn6Bf0IpDdyn4n4QalcjgnQaN7Y//j
   /Utum7tcXicPippG01rJ7Pj515NndLhCB16wCk1W15hVYuk1ZnB2U3nv+NF1n9RhhX7GS
   /Uun08b8RjqsYwDEgR2QbJjb19KITW0Wfsz8G13d0Mu9sgz6oN8aUS1myMzuw1yv720z
   /xYG1St3AQzRHIRSdKqFFjQCGzmUB+HP4mEo4AM4dp01b5SMqEF7BJYddkX6UkyfKw1a1+FDnPRSWK/Y
   /WQoFvo5bMXerY719pbUoHF8/1YqqkLC1B
   +CPLtyQpKj0SU9g8hgLzUaNPDLNOFVgNP1NN7r4KNs8hQXwuM3ipHwzebtJozj8DBL
   /LZemLBVD73R20gPMj3zxub7RjclzR1JJ1504RxEeLWQOFtW+CLOYFT14bIQtGG2XyYURHFv71YujYGpYvDj
   /zIh9htE160mhsEuEqR+MIhMTQ/1fETfQRYwmbY+ixoLuouy1BmP6bbnqXwOhqNKMMWPPD3+fYSqlkajMe5mTCE5ao="
   );
8  IRestResponse response = client.Execute(request);

```

1.3.3 Método GetHorasExtrasPendentes

Método:	GetHorasExtrasPendentes		
Funcionalidade:	Esse método será utilizado para consultar as horas extras do subordinado.		
Verbo HTTP:	GET		
Parâmetros de Entrada (JSON):			
Campo	Tipo de Entrada	Descrição	Obrigatório
Chapa	String	Chapa do subordinado	Sim

Regras de Negócio da API:

RN1) Ao realizar a requisição, será retornada a lista de horas extras pendentes do subordinado na qual sua Chapa é igual à enviado no parâmetro.



```
1 {
2   "Success": true,
3   "Message": "consulta realizada com sucesso!",
4   "Data": [
5     {
6       "Chapa": "00001544",
7       "CodUsuarioLogado": null,
8       "SenhaUsuarioLogado": null,
9       "ChapaUsuarioLogado": null,
10      "Data": "2019-01-07T00:00:00",
11      "HoraInicio": 492,
12      "HoraFim": 510,
13      "TipoExtra": "E",
14      "Validado": "0",
15      "DataReferencia": null,
16      "IdJornada": null,
17      "HoraInicioStr": "08:12",
18      "HoraFimStr": "08:30",
19      "ImagemAprovado": 0,
20      "NumeroHoras": 18,
21      "NumeroHorasStr": "00:18",
22      "Justificativa": null,
23      "Motivo": null,
24      "detalheHoraExtra": {
25        "NomeSubordinado": "ADALBERTO DE SOUSA LUIZ",
26        "CodSecaoSubordinado": "11.01.05.20.001",
27        "SecaoSubordinado": "11.01.05.20.001 - UNIDADE DE CAPITALIZAÇÃO E SERVIÇOS FINANCEIROS",
28        "BatidasDoDia": [
29          "08:12",
30          "12:42",
31          "13:44",
32          "18:19"
33        ],
34        "CodColigada": 0,
35        "ResultValidation": null
36      },
37      "MensagemRetorno": null,
38      "DataPeriodoInicio": "2019-01-02T00:00:00",
39      "DataPeriodoFim": "2019-01-08T00:00:00",
40      "CodColigada": 1,
41      "ResultValidation": {
42        "Mensagem": null,
43        "IsValid": true,
44        "Erros": []
45      }
46    }
47  ]
48 }
```

Abaixo segue exemplo de implementação (C#).



GENERATE CODE SNIPPETS
✕

C# (RestSharp) ▾
Copy to Clipboard

```

1 var client = new RestClient("http://localhost:50538/HoraExtra/GetHorasExtrasPendentes");
2 var request = new RestRequest(Method.GET);
3 request.AddHeader("cache-control", "no-cache");
4 request.AddHeader("Connection", "keep-alive");
5 request.AddHeader("content-length", "20");
6 request.AddHeader("accept-encoding", "gzip, deflate");
7 request.AddHeader("Host", "localhost:50538");
8 request.AddHeader("Postman-Token", "c8ee6a89-c686-4d46-9660-e18eea4a6d06,0a0fa4b1-346a-4625
-95b1-9cda49769ebc");
9 request.AddHeader("Cache-Control", "no-cache");|
10 request.AddHeader("Accept", "*/");
11 request.AddHeader("User-Agent", "PostmanRuntime/7.13.0");
12 request.AddHeader("Content-Type", "application/json");
13 request.AddHeader("URLAPISGOLITE", "http://www.homolog.sgolite.sebrae.com.br/PortalAPI");
14 request.AddHeader("TokenSGO", "b63DSqfp7e6ZQAuSRi/vKAdaS19R9IBuemGaY404ukDxcg+Kc5FJ1
/YwIwagfwA1F1qkJKoXX/SV74KhUyXnIVFjHdfbrBjaTzrJ4dzLq9z/hhwErMXYPFtJ+vX1kwb4m80GQG9B3
/2xm29YlWcxgA+YsCmba7S07M7Cao4yHw4H0zz9GfvRhxR4EUsRDZBYJesu57a6hGDW7iy16kw1psdw
+JAgcV6bmf5SnbBqbIA6Ug2sFDZsXcH46ieDzvRP7d9M95jZodi8UpwDD6TWh1wGGhDqkZC9gu0Ld5uzk1Vwogki3w
ya5VIUZWCrblFndAi/9kbCe/LwPOak/frxuxsKo8rLaxzInu7cbsHVXiWnY3FnrQ
+5TYIH8CWRklegL1SjnzG3skvMsV61ZAfmpftEWFewXBQ4FF0vAIlkwvYFZji
+0E7qVZrSsZ7a7CFK2018BTMBPWy9ZveUoc9tMSXxJR/SFy+6Rq+Qn6Bf0IpDdyn4n4Qa1cJgnQaN7Y//j
/Utuw7tcXicPippG01rJ7Pj515NndLhCB16wCkJW15hVYuk1ZnB2U3nv+NF1n9RhhX76S
/Uun08bBRjqsYwDEgR2QbJjb19KITW0WFsz8G13d0Mu9sgz6oN8aUS1myMzuw1yv720z
/xYG1St3AQzRHIRSDkqFFjQCGzmUB+HP4mEo4AM4dp01bSMqEF7BJYddkX6UkyYfKwAl+FDnPRSWK/Y
/WQoFvo5bMXerY719pbUoHF8/1YqqkLC1B
+CPLtyQpKj0SU9gBhgLzUaNPDLNOFVgNP1NN7r4KNs8hQXwuM3ipHWzebtJozj8DBL
/LZemLBVD73R20gPMj3zxub7RjclzR1JJ1S04RxEeLWQ0Ftw+CLOYFT14bIQtGG2XyYURHFrV71YuJGpYvDj
/zIh9htE160mhsEuEqR+MIhMTQ/1fETfqRYwmbY+ixoluouy1BmP6bbnqXw0hqNKMWPPD3+fYSqWkajMe5mTCE5ao
=");
15 request.AddParameter("undefined", "{\\"Chapa\\":\\"00001514\\"}", ParameterType.RequestBody);
16 IRestResponse response = client.Execute(request);

```

1.4 API para Aprovar Hora Extra

Esta API deverá ser utilizada pelos fornecedores para aprovação de horas extras.

1.4.1 JSON padrão retornado pela API

Esta API retornará um JSON padronizado com a seguinte estrutura:

```
{
  "Success": false,
  "Message": "",
  "Data": Objeto enviado
}
```

Os campos retornados pelo JSON são descritos na tabela abaixo.

Campo	Tipo de Retorno	Descrição
Success	True/false	Indica se a requisição foi realizada com sucesso.



Message	String	Mensagem de sucesso ou falha apresentada durante a requisição.
Data	Objeto de retorno em JSON	Dados retornados pela requisição em formato JSON

1.4.2 Header

Para consumir a API de Aprovar Hora Extra, deve ser informado no header os campos:

- **URLAPISGOLITE:** Url do Portal Api SGO Lite para autenticação. No caso do ambiente de homologação será: <http://www.homolog.sgolite.sebrae.com.br/PortalAPI/>
- **TokenSGO:** Token referente ao usuário autenticado, para obter esse Token é necessário consumir a API do PortalAPI SGOLite conforme explicado no item 1.2 deste documento



The screenshot shows the Postman interface for a POST request to `http://localhost:51174/api/NotasFiscaisTerceiro/IncluirNotaFiscal`. The 'Headers' tab is active, showing two headers:

KEY	VALUE	DESCRIPTION
URLAPISGOLITE	http://localhost:81/PortalAPI	
TokenSGO	IYTzE5cpPHHsoWi6KUKvd7fIXcHHzPA3/6X5vi+k3ECaXFqzWPrFyrlnS4DKoQQDgb+lgXuihT...	

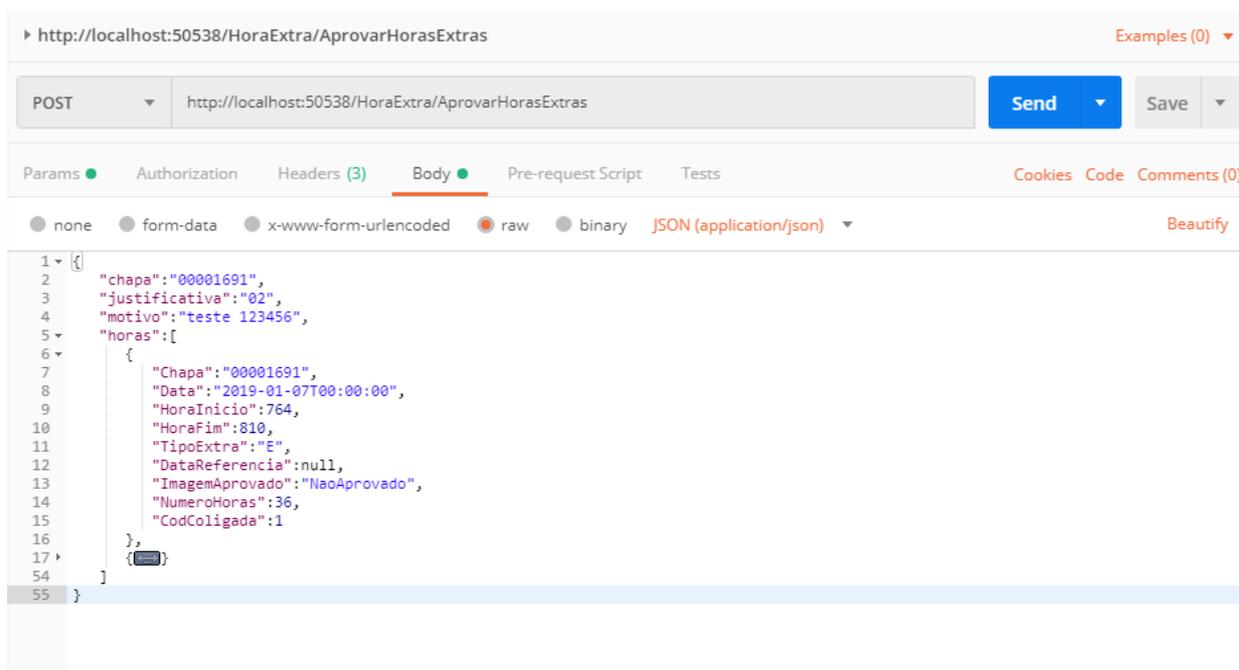
A red box highlights the two header rows, and a red arrow points to the 'URLAPISGOLITE' header key.



1.4.3 Método AprovarHorasExtras

Método:	AprovarHorasExtras				
Funcionalidade:	Esse método será utilizado para aprovação de horas extras do subordinado.				
Verbo HTTP:	POST				
Parâmetros de Entrada (JSON):					
Campo	Tipo de Entrada	Descrição	Obrigatório		
Chapa	String	Chapa do subordinado.	Não		
Motivo	String	Descrição do motivo da aprovação.	Sim		
Justificativa	String	Código da justificativa da hora extra (ex.: 01, 02, 03...)	Não		
Horas	Lista[]	Campos da lista de horas que serão aprovadas			
		Campo	Tipo de Entrada	Descrição	Obrigatório
		Chapa	String	Chapa do subordinado da hora extra	Sim
		Data	Datetime	Data da Ocorrência	Sim
		HorasInicio	Int	Quantidade da hora em que iniciou a hora extra (em minutos, ex: 75)	Sim
		HoraFim	Int	Quantidade da hora em que finalizou a hora extra (em minutos, ex: 75)	Sim
		TipoExtra	String	Tipo da ocorrência (ex: E, R, D...)	Sim
		DataReferencia	Datetime	Data de referência da ocorrência	Não
		NumeroHoras	Int	Quantidade de horas extras que será aprovado para a ocorrência (em minutos, ex: 145)	Sim
		CodColigada	Short	Código da coligada	Sim
ImagemAprovado	String	Status da ocorrência (ex: NaoAprovado, ou Aprovado)	Sim		

Para acionar o método de Aprovar horas extras, deverá ser enviado no corpo da requisição o os parâmetros citados na tabela acima no formato Json.

```

1 {
2   "chapa": "00001691",
3   "justificativa": "02",
4   "motivo": "teste 123456",
5   "horas": [
6     {
7       "Chapa": "00001691",
8       "Data": "2019-01-07T00:00:00",
9       "HoraInicio": 764,
10      "HoraFim": 810,
11      "TipoExtra": "E",
12      "DataReferencia": null,
13      "ImagemAprovado": "NaoAprovado",
14      "NumeroHoras": 36,
15      "CodColigada": 1
16    },
17  ]
18 }

```

Abaixo segue exemplo de implementação (C#).

GENERATE CODE SNIPPETS
✕

C# (RestSharp) ▾
Copy to Clipboard

```

1 var client = new RestClient("http://localhost:50538/HoraExtra
2 /AprovarHorasExtrasSubordinados");
3 var request = new RestRequest(Method.POST);
4 request.AddHeader("Postman-Token", "2321b795-6355-4162-aca3-0bd2c09b17ce");
5 request.AddHeader("cache-control", "no-cache");
6 request.AddHeader("Content-Type", "application/json");
7 request.AddHeader("URLAPISGOLITE", "http://localhost:50538/Porta1API");
8 request.AddHeader("TokenSGO", "b63DSqfp7e6ZQAuSRI/vKAdaS19R9IBuemGaY404ukDxcg+Kc5FJ1
9 /Yw1wagfwA1F1qkJKoXX/SV74KhUyXnIVfjHdfbrBjaTzrJ4dzLq9z/hhwErMXYPftJ+vX1kwb4m80GQG9B3
10 /2xm29YLWCxgA+YsCmba7S07M7Cao4yHw4H0zz9GfvRxrR4EUsRDZBYJesu57a6hGDW7iy16kwl1psdw
11 +JAgcV6bmf55nVbqBIA6Ug2sFDZzScH46ieDzvRP7d9M95jZodi8UpWDD6Twh1wGghDqkZC9gu0Ld5uzk1Vwogki
12 3wya5VIUZWCrbLFndAi/9kbCe/LwPOak/frxuxsKo8rLaxzInu7cbshVXiwnY3FnrQ
13 +5TYIH8CWRklegL1SjnzG3skvMsV612AfmPftEWFewXBQ4FF0vAI1kwvYFZji
14 +OE7qVZRssZ7a7CfK2018BTMBPWy9ZveUoc9tMSXxJR/SFy+6Rq+Qn6BFOIPddyn4n4Qa1cJgnQaN7Y//j
15 /UtuW7tcXicPippG01rJ7Pj515NndLhCB16wCk1W15hVYuk1ZnB2U3nv+NF1n9RhhX7G5
16 /Uun08bBRjqsYwDEgR2QbJjb19KITW0Wfsz8G13d0Mu9sgz6oN8aUS1myMzuw1yv720z
17 /xYG1St3AQzRHIRSdKqFFjQCGzmUB+HP4mEo4AM4dp01bSMqEF7BjYddkX6UkyfKw1+FDnPRSWK/Y
18 /WQoFvo5bMXerY719pbUoHF8/1YqqkLC1B
19 +CPLtyQpKj0SU9gBhgLzUaNPDLNOFVgNP1NN7r4KNs8hQXwuM3ipHwzbtJozj8DBL
20 /LZemLBVD73R20gPMj3zXub7Rjc1zR1JJ1S04RxEeLWQOfTW+CLOYFT14bIQGTGG2XyYURHFrV71YujYgPvYdJ
21 /zIh9htE160mhsEuEqR+MIhMTQ/1fETfqRYwmbY+ix0Luouy1BmP6bbnqXw0hQNKMWPPD3
22 +fYSqkKajMe5mTCE5ao=");
23 request.AddParameter("undefined", "{ \r\n \r\n \"chapa\": \"00001691\", \r\n \r\n \"justificativa\"
24 : \"02\", \r\n \r\n \"motivo\": \"teste 123456\", \r\n \r\n \"horas\": [ \r\n \r\n { \r\n \r\n
25 : \"CodColigada\": 1, \r\n \r\n : \"Chapa\": \"00001694\", \r\n \r\n : \"Data\": \"2019-01
26 -08T00:00:00\", \r\n \r\n : \"HoraInicio\": 735, \r\n \r\n : \"HoraFim\": 818, \r\n \r\n
27 : \"TipoExtra\": \"E\", \r\n \r\n : \"DataReferencia\": null, \r\n \r\n : \"NumeroHoras\"
28 : 12, \r\n \r\n } \r\n \r\n { \r\n \r\n : \"CodColigada\": 1, \r\n \r\n : \"Chapa\"
29 : \"00001691\", \r\n \r\n : \"Data\": \"2019-01-08T00:00:00\", \r\n \r\n : \"HoraInicio\"
30 : 795, \r\n \r\n : \"HoraFim\": 810, \r\n \r\n : \"TipoExtra\": \"E\", \r\n \r\n
31 : \"DataReferencia\": null, \r\n \r\n : \"NumeroHoras\": 15, \r\n \r\n } \r\n \r\n } \r\n \r\n }\",
32 ParameterType.RequestBody);

```



Importante: Ao enviar os dados para aprovação, é importante que todas as horas da lista horas estejam com o valor “NumeroHoras” para que seja possível realizar aprovação da hora extra com a quantidade definida.

Regras de Negócio da API:

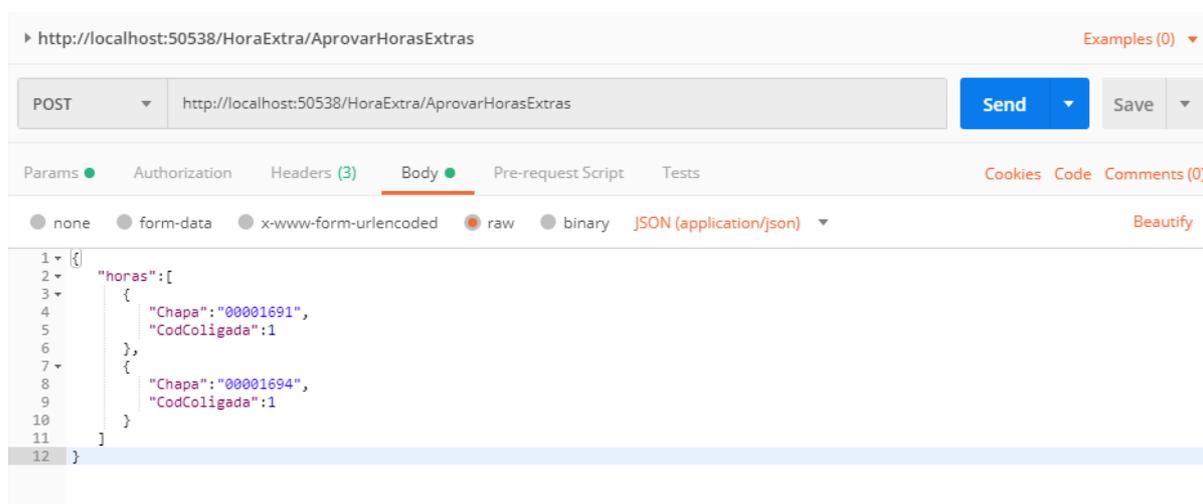
RN1) Ao realizar a requisição o webmetodo irá aprovar as horas extras daquele subordinado listadas no campo “horas”. Será considerado o número de horas enviado na lista, seja ela parcial ou total (ex: de 30 minutos de hora extra deseja-se aprovar apenas 20, então deve ser enviado no campo “NumeroHoras” o valor 20).

RN2) As validações serão realizadas pelo serviço de Aprovação de Hora Extra nativo do RM e será retornado uma string com o log caso ocorra algum problema na aprovação.

1.4.4 Método AprovarHorasExtrasSubordinados

Método:	AprovarHorasExtrasSubordinados				
Funcionalidade:	Esse método será utilizado para aprovação de horas extras dos subordinados.				
Verbo HTTP:	POST				
Parâmetros de Entrada (JSON):					
Campo	Tipo de Entrada	Descrição	Obrigatório		
Horas	Lista[]	Campos da lista de horas que serão aprovadas			
		Campo	Tipo de Entrada	Descrição	Obrigatório
		Chapa	String	Chapa do subordinado da hora extra	Sim
		CodColigada	Short	Código da coligada	Sim

Para acionar o método de Aprovar horas extras dos subordinados, deverá ser enviado no corpo da requisição os parâmetros citados na tabela acima no formato Json.



The screenshot shows a REST client interface with the following details:

- URL: `http://localhost:50538/HoraExtra/AprovarHorasExtras`
- Method: `POST`
- Body (JSON):


```

1 {
2   "horas": [
3     {
4       "Chapa": "00001691",
5       "CodColigada": 1
6     },
7     {
8       "Chapa": "00001694",
9       "CodColigada": 1
10    }
11  ]
12 }
      
```

Importante: Ao enviar os dados para aprovação, não é necessário enviar os campos referentes à hora, pois o webmétodo irá considerar apenas a chapa e o codColigada do subordinado.



Regras de Negócio da API:

RN1) O webmétodo irá buscar as horas extras de cada subordinado da lista. Após isso será aprovada **TODAS** as horas extras dos subordinados listados considerando a hora extra total.

RN2) As validações serão realizadas pelo serviço de Aprovação de Hora Extra nativo do RM e será retornado uma string com o log caso ocorra algum problema na aprovação.

Abaixo segue exemplo de implementação (C#).

```
GENERATE CODE SNIPPETS ✕
```

```
C# (RestSharp) Copy to Clipboard
```

```
1 var client = new RestClient("http://localhost:50538/HoraExtra/AprovarHorasExtras");
2 var request = new RestRequest(Method.POST);
3 request.AddHeader("Postman-Token", "b9fa3f27-8ba4-4c01-a7fd-74748cd65606");
4 request.AddHeader("cache-control", "no-cache");
5 request.AddHeader("Content-Type", "application/json");
6 request.AddHeader("URLAPISGOLITE", "http://localhost:50538/PortalAPI");
7 request.AddHeader("TokenSGO", "b63DSqfp7e6ZQAuSRi/vKAdaS19R9IBuemGaY404ukDxcg+Kc5FJ1
  /Yw1wagfwA1F1qkJkoXX/SV74KhUyXnIVfjHdfbrBjaTzrJ4dzLq9z/hhwErMXYPftJ+vX1kwb4m80GQG9B3
  /2xm29YLWCxgA+YsCmba7S07M7Cao4yHw4H0zz9GfvrXhR4EUsRDZBYJesu57a6hGDW7iy16kwl1psdw
  +JAgcV6bmf55SnbqBIA6Ug2sFDZxScH46ieDzvRP7d9M95jZodi8UpWDD6TWh1wGGhDqkZC9gu0Ld5uzk1VWogki3w
  ya5VIUZWCrblFndAi/9kbCe/LwPOak/frxuxsKo8rLaxZInu7cbshVXiWnY3FnrQ
  +5TYIH8CWRklegL15jnzG3skvMsV61ZAfmpftEWFewX8Q4FF0vAI1kwvYFZJi
  +OE7qVZRssZ7a7CfK2018BTMBPWy9ZveUoc9tMSXxJR/SFy+6Rq+Qn68FOIpDdyn4n4Qa1cJgnQaN7Y//j
  /UtuW7tcXicPippG01rJ7Pj515NndLhCB16wCk1W15hVYuk1Zn82U3nv+NF1n9RhhX7GS
  /Uun08bBRjqsYwDEgR2QbJjb19KITW0Wfsz8G13d0Mu9sgz6oN8aUS1myMzuw1yv720z
  /xYG1St3AQzRHIRsdKqFFjQCGzmUB+HP4mEo4AM4dp01bSMqEF7BJYddkX6UkyfFKWa1+FDnPRSWK/Y
  /WQoFvo5bMXerY719pbUoHF8/1YqqkLC1B
  +CPLtyQpKj0SU9g8hgLzUaNPDLNOFvgNP1NN7r4KNs8hQXwuM3ipHMzebtJozj8DBL
  /LZemLBVD73R20gPMj3zXub7RjclzR1JJ1S04RxEeLWQOfTW+CLOYFT14bIQtGG2XyYURHFv71YuYjYgPvYDj
  /zIh9htE160mhsEuEqR+MIhMTQ/1fETfqRYwmbY+ixoLuouy18mP6bbnqXwOhqNKMwPPD3+fYSqikajMe5mTCE5ao
  =");
8 request.AddParameter("undefined", "{\r\n  \"horas\":[ \r\n    { \r\n      \"Chapa\"
  :\"00001691\", \r\n      \"CodColigada\":1\r\n    }, \r\n    { \r\n
  \"Chapa\": \"00001694\", \r\n      \"CodColigada\":1\r\n    } \r\n  ]\r\n}",
  ParameterType.RequestBody);
9 IRestResponse response = client.Execute(request);
```



1.5 API para consultar Justificativas

Esta API deverá ser utilizada pelos fornecedores para consulta de justificativas de hora extra.

1.5.1 JSON padrão retornado pela API

Esta API retornará um JSON padronizado com a seguinte estrutura:

```
{
  "Success": false,
  "Message": "",
  "Data": Objeto enviado
}
```

Os campos retornados pelo JSON são descritos na tabela abaixo.

Campo	Tipo de Retorno	Descrição
Success	True/false	Indica se a requisição foi realizada com sucesso.
Message	String	Mensagem de sucesso ou falha apresentada durante a requisição.
Data	Objeto de retorno em JSON	Dados retornados pela requisição em formato JSON

1.5.2 Header

Para consumir a API de Aprovar Hora Extra, deve ser informado no header os campos:

- **URLAPISGOLITE:** Url do Portal Api SGO Lite para autenticação. No caso do ambiente de homologação será: <http://www.homolog.sgolite.sebrae.com.br/PortalAPI/>
- **TokenSGO:** Token referente ao usuário autenticado, para obter esse Token é necessário consumir a API do PortalAPI SGOLite conforme explicado no item 1.2 deste documento



The screenshot shows a REST client interface with the following details:

- Method: POST
- URL: http://localhost:51174/api/NotasFiscaisTerceiro/IncluirNotaFiscal
- Headers (2):

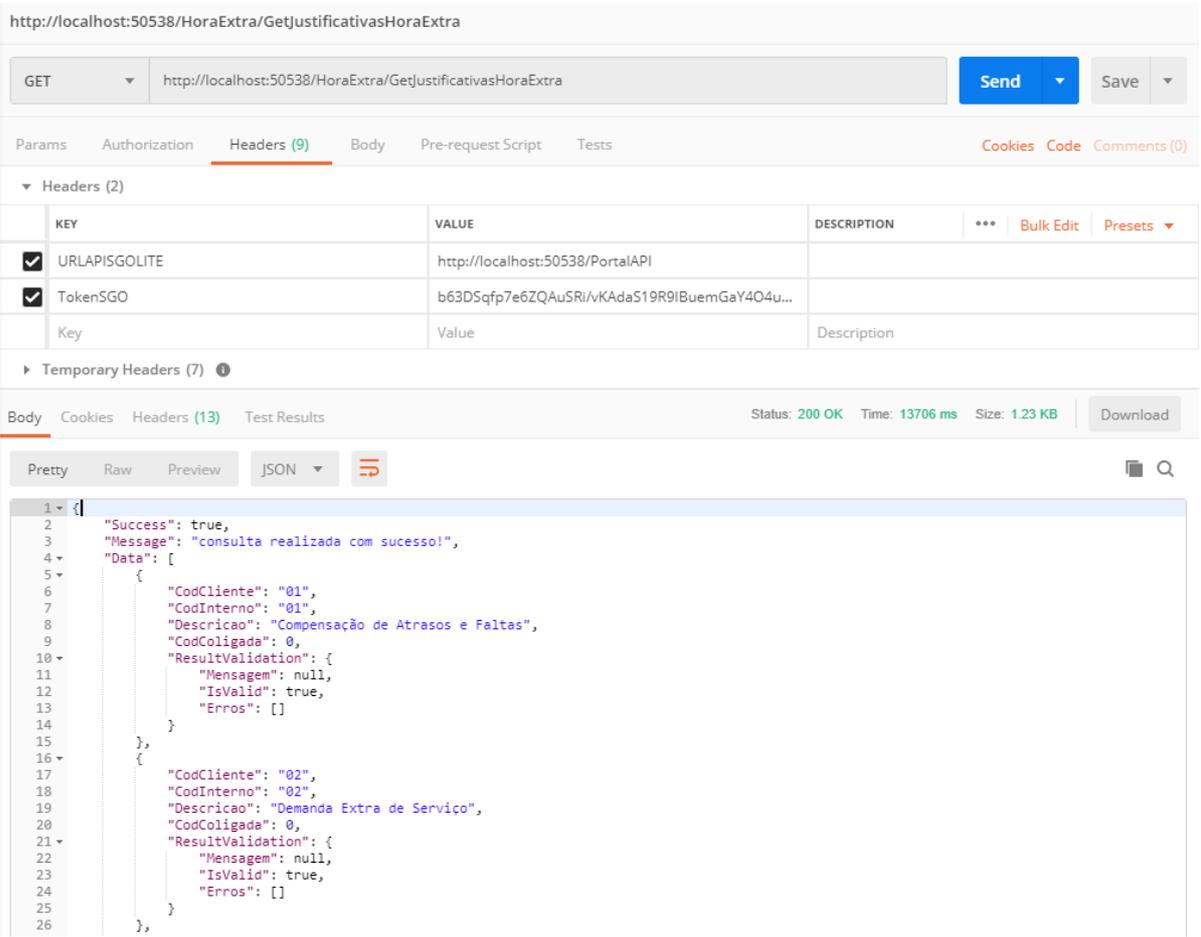
KEY	VALUE	DESCRIPTION
URLAPISGOLITE	http://localhost:81/PortalAPI	
TokenSGO	IYTzE5cpPHHsoWi6KUKvd7fIXcHzPA3/6X5v+k3ECaXFqzWPrFytInS4DKoQDgb+lgXuihT...	



1.5.3 Método GetJustificativasHoraExtra

Método:	GetJustificativasHoraExtra		
Funcionalidade:	Esse método será utilizado para consultar as justificativas de hora extra.		
Verbo HTTP:	GET		
Parâmetros de Entrada (JSON):			
Campo	Tipo de Entrada	Descrição	Obrigatório
		Não se aplica	

Para acionar o método de Aprovar horas extras dos subordinados, deverá ser enviado no corpo da requisição os parâmetros citados na tabela acima no formato Json.



The screenshot shows a REST client interface with the following details:

- URL:** http://localhost:50538/HoraExtra/GetJustificativasHoraExtra
- Method:** GET
- Headers:**
 - URLAPISGOLITE: http://localhost:50538/PortalAPI
 - TokenSGO: b63DSqfp7e6ZQAuSRi/vKAdaS19R9IBuemGaY4O4u...
- Status:** 200 OK, Time: 13706 ms, Size: 1.23 KB
- Response Body (JSON):**

```

1- {
2-   "Success": true,
3-   "Message": "consulta realizada com sucesso!",
4-   "Data": [
5-     {
6-       "CodCliente": "01",
7-       "CodInterno": "01",
8-       "Descricao": "Compensação de Atrasos e Faltas",
9-       "CodColigada": 0,
10-      "ResultValidation": {
11-        "Mensagem": null,
12-        "IsValid": true,
13-        "Erros": []
14-      }
15-    },
16-    {
17-      "CodCliente": "02",
18-      "CodInterno": "02",
19-      "Descricao": "Demanda Extra de Serviço",
20-      "CodColigada": 0,
21-      "ResultValidation": {
22-        "Mensagem": null,
23-        "IsValid": true,
24-        "Erros": []
25-      }
26-    }
27-  ]
28- }

```



Abaixo segue exemplo de implementação (C#).

```
GENERATE CODE SNIPPETS ✕
```

C# (RestSharp) Copy to Clipboard

```
1 var client = new RestClient("http://localhost:50538/HoraExtra/GetJustificativasHoraExtra");
2 var request = new RestRequest(Method.GET);
3 request.AddHeader("cache-control", "no-cache");
4 request.AddHeader("Connection", "keep-alive");
5 request.AddHeader("accept-encoding", "gzip, deflate");
6 request.AddHeader("Host", "localhost:50538");
7 request.AddHeader("Postman-Token", "e3c51577-4882-4b06-a153-42d84596fc78,279bec19-a5aa-4818-81a9-1d886bd16b1e");
8 request.AddHeader("Cache-Control", "no-cache");
9 request.AddHeader("Accept", "*/");
10 request.AddHeader("User-Agent", "PostmanRuntime/7.13.0");
11 request.AddHeader("TokenSG0", "b63DSqfp7e6ZQAuSRi/vKAdaS19R9IBuemGaY404ukDxcg+Kc5FJ1
/Yw1wagfwA1F1qkJkoXX/SV74KhUyXnIVFjHdfbrBjaTzrJ4dzLq9z/hhwErMXYPFtJ+vX1kLb4m80GQG983
/2xm29YLWCxgA+YsCmba7SO7M7Cao4yHw4H0zz9GfvRhxR4EUsRDZBYJesu57a6hGDW7iy16kw1psdw
+JAgcV6bmf5Snb8qbIA6Ug2sFDZxSCH46ieDzvRP7d9M95jZodi8UpWDD6Twh1wGGhDqkZC9gu0Ld5uzk1Vwogki3w
ya5VIUZWCrbLFndAi/9kbCe/LwPOak/frxuxsKo8rLAXzInu7cbsHVXiWnY3FnrQ
+5TYIH8CWRk1egL1SjnzG3skvMsV61ZAfmpftEWFewXBQ4FF0vAI1kwvYFZJi
+OE7qVZRssZ7a7CfK2018BTMBPwy9ZveUoc9tMSXxJR/SFy+6Rq+Qn68f0IpDdyn4n4Qa1cJgnQaNY//j
/UtuW7tcXicPippG01rJ7Pj515NndLhCB16wCkJW15hVYuk1Zn82U3nv+NF1n9RhhX7G5
/UunO8bBRjqsYwDEgR2QbJjb19KITW0Wfsz8G13d0Mu9sgz6oN8aUS1myMzuw1yv720z
/xYG1St3AQzRHIRSdKqFFjQCGzmUB+HP4mEo4AM4dp01b5MqEF7B7YddkX6UkyfKw1a1+FDnPRSWK/Y
/WQoFvo5bMXerY719pbUoHF8/1YqqkLC1B
+CPLtyQpKj0SU9g8hgLzUaNPDLNOFVgNP1NN7r4KNs8hQXwuM3ipHwzbtJozj8DBL
/LZemLBVD73R20gPMj3zXub7RjclzR1JJ1S04RxEeLWQ0FtW+CLOYFT14bIQtGG2XyYURHFv71YuYjGpYvDj
/zIh9htE160mhsEuEqR+MIhMTQ/1fETfQRYwmbY+ix0Luouy1BmP6bbnqXwOhqNKMwPPD3+fYSqwkajMe5mTCE5ao
=");
12 request.AddHeader("URLAPISGOLITE", "http://localhost:50538/PortalAPI");
13 IRestResponse response = client.Execute(request);
```



1.6 Histórico de Atualização do Documento

1.6.1 Histórico de Versões

Autor	Versão	Data	Descrição das Alterações
Diego Alves Torres	1.0	03/01/2019	Criação do documento.