

# Documentação API para Sistema de Eventos

## Instruções para uso da API

Esta é uma API desenvolvida para proporcionar o gerenciamento de eventos. A API ainda está em fase de desenvolvimento, por isso, o banco será disponibilizado com um conjunto de dados para teste.

Dentre o conjunto de dados disponibilizado, há dados de autenticação dos diferentes usuários, com diferentes papéis (Administrador Geral, Administrador de Evento, Avaliador, Autor e Participante). Cada papel tem permissões específicas dentro do sistema, ou seja, acessa rotas específicas. Para acessar as rotas da aplicação, é necessário que seja informado via cabeçalho da requisição (Header), o token de acesso (gerado ao acessar a rota 'POST\_authentication') no campo "token-authorization". Os usuários previamente cadastrados no BD são:

**usuário:** teste1

**email:** email1

**senha:** 123

**papéis:** Administrador Geral, Administrador de Evento (Eventos 1 e 3), Avaliador(Evento 1), Autor(Evento 1), Participante(Evento 1)

**usuário:** teste2

**email:** email2

**senha:** 321

**papéis:** Participante(Eventos 1, 2 e 3)

**usuário:** teste3

**email:** email3

**senha:** 1234

**papéis:** Administrador de Evento (Evento 2), Avaliador(Evento 1), Autor(Evento 1), Participante(Evento 2)

## Group Users

Seção: autenticação de Usuário[authentication]

Um usuário loga no sistema[POST]

- Request (application/json)

```
{
  "email": "email1",
  "password": "123"
}
```

- Response 200(application/json)

```
{
  "login": true,
  "isAdmGeral": true,
}
```

```
"id": "1",
"name": "teste1",
"isAdmEvento": true,
"access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOiJlbnV4LnBocCI6MTQ3NzUyMjQ5MywibmJmljoxNDc3MzQyMDkyLCJkYXRhIjp7ImkljoiMSIsIm5hbWUiOiJNYWNpZWwgU291c2EiLCJlbWFpbCI6Im1hY2IibHNvdXNhQGdtYWlsLmNvbSI6ImZlZWRtR2V5YWwiOnRydWUsImV2ZW50cyI6bnVsbH19.xo35aYNcZKgF02pruW_Jx4qCZsccz5Aj-GVuDn-5XGw"}
```

- Response 401 (application/json)

```
{ "message": "Login não cadastrado!" }
```

- Response 400 (application/json)

```
{ "message": "A senha digitada está incorreta!" }
```

OR

```
{ "message": "Email preenchido de forma incorreta! O email não pode ser nulo e nem conter mais de 50 digitos." }
```

Seção: criar conta no sistema [account]

Rota utilizada para se cadastrar no sistema[POST]

- Request (application/json)

```
{
  "nome": "teste1",
  "email": "email1",
  "senha": "123",
  "sexo": "M",
  "dataDeNascimento": "2013-09-10",
  "cepEndereco": "12345678",
  "logradouroEndereco": "Endereço Teste1",
  "complementoEndereco": "Endereço Teste1",
  "idCidade": "1",
  "imagem": "fotoTeste1.jpg"
}
```

- Response 200(application/json)

```
{
  "data": {
    "UsuarioAdicionado": {
```

```

    "Id": "1",
    "Nome": "teste1",
    "Email": "email1",
    "Senha": null,
    "Sexo": "M",
    "DataDeNascimento": "2013-09-10",
    "IdEndereco": "1",
    "Imagem": "fotoTeste1.jpg"
  },
  "message": "Usuário cadastrado com sucesso!"
}

```

- Response 400 (application/json)

```

{ "message": "Email preenchido de forma incorreta! O email não pode ser nulo e nem conter mais de 50 dígitos." }

```

OR

```

{ "message": "Nome preenchido de forma incorreta! O nome não pode ser nulo e nem conter mais de 200 dígitos." }

```

OR

```

{ "message": "O CEP deve possuir exatamente 8 dígitos" }

```

- Response 404 (application/json)

```

{ "message": "Id da cidade não pode ser nulo " }

```

OR

```

{ "message": "Cidade com id 1 não existe no sistema" }

```

OR

```

{ "message": "Id de Estado não existe" }

```

OR

```

{ "message": "O campo sexo deve conter apenas um dígito! E deve ser "M" ou "F" " }

```

- Response 500 (application/json)

```

{ "message": "Mensagem de erro proveniente do banco de dados" }

```

## Seção cadastrar Usuários [users]

### Cadastrar um novo Usuário[POST]

Diferentemente da rota [POST][account], esta rota só pode ser acessada por Administrador Geral e permite que ele adicione outros Administradores Gerais.

- Papéis Com Permissão

Administrador Geral

- Request (application/json)

```
{
  "nome":"teste1",
  "email":"email1",
  "senha":"123",
  "sexo":"M",
  "dataDeNascimento":"2013-09-10",
  "cepEndereco":"23232323",
  "logradouroEndereco":"Endereço Teste1",
  "isAdmGeral":"true",
  "complementoEndereco":"Endereço Teste1",
  "idCidade":"1",
  "imagem": "fotoTeste1.jpg"
}
```

- Response 200(application/json)

```
{
  "data": {
    "UsuarioAdicionado": {
      "Id": "1",
      "Nome": "teste1",
      "Email": "email1",
      "Senha": null,
      "Sexo": "M",
      "DataDeNascimento": "2013-09-10",
      "IdEndereco": "1",
      "Imagem": "fotoTeste1.jpg"
    }
  },
  "message": "Usuário cadastrado com sucesso!"
}
```

- Response 403 (application/json)

```
{ "message": "Usuário não possui permissão" }
```

- Response 400 (application/json)

```
{ "message": "Email preenchido de forma incorreta! O email não pode ser nulo e nem conter mais de 50 dígitos." }
```

OR

```
{ "message": "Nome preenchido de forma incorreta! O nome não pode ser nulo e nem conter mais de 200 dígitos." }
```

OR

```
{ "message": "O CEP deve possuir exatamente 8 dígitos" }
```

- Response 500 (application/json)

```
{ "message": "Mensagem de erro proveniente do banco de dados" }
```

## Seção: Usuários [users]

### Lista de Usuários [GET]

Busca todos os Usuários cadastrados.

Retorna uma lista com todos os usuários cadastrados no sistema

- Papéis Com Permissão

Administrador Geral

- Response 200 (application/json)

```
{
  "data": [
    {
      "Id": "1",
      "Nome": "teste1",
      "Email": "email1",
      "Senha": null,
      "Sexo": "M",
      "DataDeNascimento": "2013-09-10",
      "IdEndereco": "1",
      "Imagem": null
    },
    {
      "Id": "2",
      "Nome": "teste2",
      "Email": "email2",

```

```

    "Senha": null,
    "Sexo": "F",
    "DataDeNascimento": "1990-10-05",
    "IdEndereco": "2",
    "Imagem": null
  }
],
"message": "Usuários buscados com sucesso!"}

```

- Response 400 (application/json)  
{"message": "Erro ao buscar usuários! Desculpe o transtorno, trabalharemos para identificar tal erro"}
- Response 403 (application/json)  
{ "message": "Usuário não possui permissão"}
- Response 500 (application/json)  
{"message": "Mensagem de erro proveniente do banco de dados" }

Seção: busca um usuário específico [/users/{idUser}]

Retorna um usuário[GET]

Mostra um usuário específico, através de seu identificador.

- Papéis Com Permissão  
Administrador Geral
- Parameters
  - idUser : 1 (number) - Um identificador único do Usuário

- Response 200(application/json)

```

{
  "data": {
    "Id": "2",
    "Nome": "teste2",
    "Email": "email2",
    "Senha": null,
    "Sexo": "F",
    "DataDeNascimento": "1990-10-05",
    "IdEndereco": "2",
    "Imagem": null
  },
  "message": "Usuário buscado com sucesso!"}

```

- Response 400 (application/json)  
{"message": "Erro ao buscar usuários! Desculpe o transtorno, trabalharemos para identificar tal erro"}
- Response 403 (application/json)  
{ "message": "Usuário não possui permissão" }
- Response 404 (application/json)  
{"message": "Id de usuário não existe"}
- Response 500 (application/json)  
{"message": "Mensagem de erro proveniente do banco de dados" }

Seção: altera perfil de um usuário [/users/{idUser}/profile]

Altera dados de um usuário[PATCH]

- Papéis Com Permissão  
Usuário
- Parameters
  - idUser : 1 (number) - Um identificador único do Usuário
- Request(application/json)
 

```
{
  "nome": "teste12",
  "email": "email12",
  "sexo": "M",
  "dtNascimento": "2013-09-10",
  "cepEndereco": "23232323",
  "logradouroEndereco": "Endereço 1",
  "complementoEndereco": "Endereço 1",
  "idCidade": "1"
}
```

- Response 200(application/json)

```
{
  "data": [
    {
      "Id": "1",
      "Nome": "teste1",
      "Email": "email1",
      "Senha": null,
      "Sexo": "M",
      "DataDeNascimento": "2013-09-10",
      "IdEndereco": "1",
    }
  ],
  "message": "Perfil do usuário editado com sucesso!"
}
```

- Response 403 (application/json)

```
{ "message": "Usuário não possui permissão" }
```

- Response 400 (application/json)

```
{ "message": "Erro ao editar perfil! Desculpe o transtorno, trabalharemos para identificar tal erro." }
```

OR

```
{ "message": "Email preenchido de forma incorreta! O email não pode ser nulo e nem conter mais de 50 dígitos." }
```

- Response 404 (application/json)

```
{ "message": "Id de usuário não existe" }
```

- Response 500 (application/json)

```
{ "message": "Mensagem de erro proveniente do banco de dados" }
```

OR

```
{ "message": "O email informado já pertence a um usuário! Por favor, modifique o email!" }
```



Seção: altera senha de um usuário [/users/{idUser}/password]

Altera senha de um usuário[PATCH]

- Papéis Com Permissão
- Usuário
- Parameters
  - idUser : 1 (number) - Um identificador único do Usuário
- Request (application/json)

```
{  
  "password": "321"  
}
```

- Response 200 (application/json)

```
{  
  "data": {  
    "UsuarioEditado": {  
      "Id": "1",  
      "Nome": "teste1",  
      "Email": "email1",  
      "Senha": null,  
      "Sexo": "M",  
      "DataDeNascimento": "2013-09-10",  
      "IdEndereco": "1",  
      "Imagem": null  
    }  
  },  
  "message": "Senha alterada com sucesso!"  
}
```

- Response 403 (application/json)  

```
{ "message": "Usuário não possui permissão" }
```
- Response 400 (application/json)  

```
{ "message": "O campo senha não pode ser nulo!" }
```
- Response 404 (application/json)  

```
{ "message": "Id de usuário não existe!" }
```

- Response 500 (application/json)

{ "message": "Mensagem de erro proveniente do banco de dados" }

Seção: apaga um usuário específico [/users/{idUser}]

Retorna um usuário[DELETE]

Apaga um usuário específico, através de seu identificador.

- Papéis Com Permissão

Administrador Geral

- Parameters

- idUser : 1 (number) - Um identificador único do Usuário

- Response 200(application/json)

{ 2 , "message": "Usuário deletado com sucesso!" }

- Response 403 (application/json)

{ "message": "Usuário não possui permissão" }

- Response 400 (application/json)

{ "message": "Erro ao excluir usuário! Desculpe o transtorno, trabalharemos para identificar tal erro." }

- Response 404 (application/json)

{ "message": "Id de usuário não existe!" }

- Response 500 (application/json)

{ "message": "Mensagem de erro proveniente do banco de dados" }

## Group Events

Seção: Eventos [/events]

## Cadastrar um novo Evento[POST]

Rota selecionada para que o usuário possa cadastrar evento.

- Papéis Com Permissão

## Todos Usuários

- Request (application/json)

```
{
  "nome": "Evento Teste1",
  "dtIni": "2016-11-25",
  "dtFim": "2016-12-25",
  "sigla": "EVT1",
  "descricao": "Esse é um evento para testes",
  "cepEndereco": "23232323",
  "logradouroEndereco": "Endereço 1",
  "complementoEndereco": "Endereço 1",
  "idCidade": "1"
}
```

- Response 200(application/json)

```
{ "data":  
  
{ "new_token":  
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE0Nzg3MTU0MzYsImZcyL6I  
mZhYnJpY2EudWxiciEtdG8uYnJcL3Npc3RlbnWEtZXZlbnRvc1wvYmFja2VuZFwvY  
XBpXC9pbmRleC5waHAiLCJleHAiOiE0Nzg3MzAyMzYsIm5iZil6MTQ3ODcxNTgzN  
SwiZGF0YSI6eyJpZCI6IjMiLCJuYW1lIjoiaGVzdGUzliwiZW1haWwiOiJlbWFpbDMiL  
CJpc0FkbUdlcmFsIjpmYWxzZSwiZXZlbnRzIjpbeJJZEVE2ZW50byl6IjEiLCJOe21IRX  
ZlbnRvljoiRXZlbnRvIFRlc3RIMSIslk5vbWVQYXBlblBlaXMiOlsiQXZhbGlhZG9yIiwiaXV0  
b3liXX0seyJJZEVE2ZW50byl6IjEiLCJOe21IRXZlbnRvljoiRXZlbnRvIFRlc3RIMIslk5vb  
WVQYXBlblBlaXMiOlsiQWRtaW5pc3RyYWVvcilSLBhcnRpY2lwYW50ZSJDfSx7IkklRXZI  
bnRvljoiNCIsIklk5vbWVFdmVudG8iOiJFdmdVudG8gVG9vdGUzliwiTm9tZVBhcGVpcyl  
6WyJBZG1pbmlzdHJhZG9yll19XX19.t16fwzbfsqtm45l0w9IGdmOaFMmSLY_ogsHAr  
n-Vbnw",  
  
"idEventoCriado": "4" },  
  
"message": "Evento criado com sucesso!" }
```

- Response 400 (application/json)

```
{"message": "Nome do Evento não pode ser nulo"}
```

```
{"message": "Datas não podem ser nulas!"}
```

```
{"message": "O CEP deve possuir exatamente 8 digitos!"}
```

- ```
{"message": "O Id da cidade não pode ser nulo"}
```

```
{"message": "Cidade com id '99999' não existe no sistema!"}
```

- ```
{"message": "Mensagem de erro proveniente do banco de dados" }
```

Rota pela qual o usuário seleciona para participar de um Evento

- **idEvent** : 1 (number) - Um identificador único do Evento

- [illegible]

6WyJQYXJ0aWNpcGFudGUlXX0seyJJZEV2ZW50byl6ljQiLCJOb21lRXZlbnRvIjoiR  
XZlbnRvIFRlc3RlMSIsIk5vbWVQYXBlaXMiOlsiQWRtaW5pc3RyYWRvciJdfV19fQ.U  
hTNdOOO\_C-wKJUBXpwwKJKd24Nse28J8jQ2z4rh0Eo",

"joined": true },

"message": "Inscrição realizada com sucesso!" }

- Response 400 (application/json)

{"message": "Usuário já é participante deste evento!" }

- Response 404 (application/json)

{"message": "Id de evento não existe!" }

- Response 500 (application/json)

{"message": "Mensagem de erro proveniente do banco de dados" }

Seção: busca todos os Eventos [events]

Lista de Eventos [GET]

Eu como administrador geral eu quero ver a lista de todos os eventos.

- Papéis Com Permissão

Todos Usuários

- Response 200 (application/json)

```
{
  "data": [
    {
      "Id": "1",
      "Nome": "Evento Teste1",
      "Sigla": "EVT1",
      "Descricao": "Este é um evento para testes",
      "IdEndereco": "1",
      "DtInicio": "2016-11-25",
      "DtFinal": "2016-12-25"
    },
    {
      "Id": "2",
      "Nome": "Evento Teste1",
      "Sigla": "EVT2",
      "Descricao": "Este é um evento para testes",
      "IdEndereco": "1",

```

```

        "DtInicio": "2016-11-25",
        "DtFinal": "2016-12-25"
    },
    {
        "Id": "3",
        "Nome": "Evento Teste1",
        "Sigla": "EVT3",
        "Descricao": "Este é um evento para testes",
        "IdEndereco": "3",
        "DtInicio": "2016-11-25",
        "DtFinal": "2016--12-25"
    }
],
    "message": "Eventos buscados com sucesso!"
}

```

- Response 400 (application/json)

```

{"message": "Erro ao buscar eventos! Desculpe o transtorno, trabalharemos para identificar tal erro." }

```

- Response 500 (application/json)

```

{"message": "Mensagem de erro proveniente do banco de dados" }

```

Seção: busca todos os Eventos [events/adm-by-me]

Lista de Eventos [GET]

Eu como administrador quero listar todos os evento que administro.

- Papéis Com Permissão

Administrador de Evento

- Response 200 (application/json)

```

{
  "data": [
    {
      "Id": "1",
      "Nome": "Evento Teste1",
      "Sigla": "EVT1",
      "Descricao": "Este é um evento para testes",
      "IdEndereco": "1",
      "DtInicio": "2016-11-25",
      "DtFinal": "2016-12-25"
    }
  ]
}

```

```

    },
    {
      "Id": "2",
      "Nome": "Evento Teste2",
      "Sigla": "EVT2",
      "Descricao": "Este é um evento para testes",
      "IdEndereco": "1",
      "DtInicio": "2015-10-20",
      "DtFinal": "2015-11-20"
    }
  ],
  "message": "Eventos buscados com sucesso!"
}

```

- Response 403 (application/json)
 

```
{ "message": "Usuário não possui permissão" }
```
- Response 400 (application/json)
 

```
{ "message": "Erro ao buscar eventos! Desculpe o transtorno, trabalharemos para identificar tal erro." }
```
- Response 500 (application/json)
 

```
{ "message": "Mensagem de erro proveniente do banco de dados" }
```

Seção: busca um evento [events/{idEvent}]

Busca um evento específico [GET]

- Papéis Com Permissão
  - Todos Usuários
- Parameters
  - idEvent : 1 (number) - Um identificador único do Evento
- Response 200 (application/json)

```

{
  "data": [
    {
      "Id": "1",
      "Nome": "Evento Teste1",
      "Sigla": "EVT1",

```

```

      "Descricao": "Este é um vento para testes",
      "IdEndereco": "1",
      "DtInicio": "2019-02-02",
      "DtFinal": "2019-03-03"
    }
  ],
  "message": "Evento buscado com sucesso!"
}

```

- Response 400 (application/json)

```

{"message": "Erro ao buscar eventos! Desculpe o transtorno, trabalharemos para identificar tal erro." }

```

- Response 404 (application/json)

```

{"message": "Id de evento não existe!" }

```

- Response 500 (application/json)

```

{"message": "Mensagem de erro proveniente do banco de dados" }

```

Seção: alterar um evento específico [events/{idEvent}]

Alterar dados de um Evento[PUT]

Como administrador de evento eu quero alterar os dados de um evento que administro.

- Papéis Com Permissão

Administrador do Evento

- Parameters

- idEvent : 1 (number) - Um identificador único do Usuário

- Request (application/json)

```

{
  "nome":"Evento Teste3",
  "dtInicio":"2016-11-21",
  "dtFim":"2016-12-21",
  "sigla":"EVT3",
  "descricao":"Este é um evento para testes",
  "cepEndereco":"23232323",
  "logradouroEndereco":"Endereço 3",

```



```
"complementoEndereco":"Endereço 3!",  
"idCidade":"3"  
}
```

- Response 200(application/json)

```
{  
  "data": {  
    "EventoEditado": {  
      "Id": "4",  
      "nome":"Evento Teste3",  
      "dtInicio":"2016-11-21",  
      "dtFim":"2016-12-21",  
      "sigla":"EVT3",  
      "descricao":"Este é um evento para testes",  
    }  
  },  
  "message": "Evento alterado com sucesso!"  
}
```

- Response 403 (application/json)

```
{ "message": "Usuário não possui permissão"}
```

- Response 400 (application/json)

```
{ "message": "O CEP deve possuir exatamente 8 dígitos!"}
```

OR

```
{ "message": "Nome do evento não pode ser nulo!"}
```

OR

```
{ "message": "Datas não podem ser nulas!"}
```

OR

```
{ "message": "Erro ao editar Evento! Desculpe o transtorno, trabalharemos para  
identificar tal erro."}
```

- Response 404 (application/json)

```
{ "message": "Id de evento não existe!"}
```

OR

```
{ "message": "O Id da cidade não pode ser nulo"}
```

OR

```
{ "message": "Cidade com id '99999' não existe no sistema!"}
```

OR

```
{ "message": "Id de endereço não existe"}
```

- Response 500 (application/json)

```
{"message": "Mensagem de erro proveniente do banco de dados" }
```

Seção: deleta um evento específico [users/{idEvento}]

Deleta um evento[DELETE]

Deleta um evento específico.

- Papéis Com Permissão

Administrador Geral

- Parameters

- idEvento : 1 (number) - Um identificador único do Usuário

- Response 200(application/json)

```
{"data":{"IdEventoDeletado":"4"},"message":"Evento deletado com sucesso!"}
```

- Response 403 (application/json)

```
{ "message": "Usuário não possui permissão"}
```

- Response 400 (application/json)

```
{ "message": "Erro ao deletar Evento! Desculpe o transtorno, trabalharemos para identificar tal erro."}
```

OR

```
{ "message": "Evento não pode ser deletado pois há outros usuários vinculados a ele!"}
```

- Response 404 (application/json)  

```
{ "message": "Id de evento não existe!" }
```
- Response 500 (application/json)  

```
{ "message": "Mensagem de erro proveniente do banco de dados" }
```

## Group Schedule

Seção: Cronograma de um evento [events/{idEvent}/schedule]

Adiciona o cronograma de um evento [POST]

- Parameters
  - idEvent : 1 (number) - Um identificador único do Evento
- Papéis Com Permissão  
Administrador do Evento
- Request (application/json)  

```
{
  "dataLimiteSubmissao":"2015-01-11",
  "dataLimiteRevisao":"2015-02-11",
  "dataLimiteInscricao":"2015-03-11"
}
```
- Response 200 (application/json)  

```
{ "data": [
  "CronogramaAdicionado":
  { "DataLimiteInscricao":"2015-03-11", "DataLimiteSubmissao":"2015-01-11", "DataLimiteRevisao":"2015-02-11", "IdEvento":"1" } ],
  "message": "Cronograma adicionado com sucesso!" }
```
- Response 400 (application/json)  

```
{ "message": "Erro ao adicionar Cronograma! Desculpe o transtorno, trabalharemos para identificar tal erro." }
```

  
OR  

```
{ "message": "Erro ao adicionar Cronograma! Desculpe o transtorno, trabalharemos para identificar tal erro." }
```

- Response 404 (application/json)  
{"message": "Id de evento não existe!"}
- Response 500 (application/json)  
{"message": "Este evento já possui cronograma" }  
  
OR  
{"message": "Mensagem de erro proveniente do banco de dados" }

#### Retorna o cronograma de um evento [GET]

- Papéis Com Permissão  
Todos Usuários
- Response 200 (application/json)  
  
{"data": [  
{"DataLimiteInscricao": "2016-11-09", "DataLimiteSubmissao": "2016-10-09", "DataLimiteRevisao": "2016-11-05", "IdEvento": "1"} ],  
"message": "Cronograma buscado com sucesso!"}
- Response 400 (application/json)  
  
{"message": "Erro ao buscar cronograma! Desculpe o transtorno, trabalharemos para identificar tal erro."}
- Response 404 (application/json)  
  
{"message": "Id de evento não existe!"}
- Response 500 (application/json)  
  
{"message": "Mensagem de erro proveniente do banco de dados" }

#### Altera o cronograma de um evento [PUT]

- Papéis Com Permissão  
Administrador de Evento

- Request (application/json)

```
{
  "dataLimiteSubmissao":"2016-01-11",
  "dataLimiteRevisao":"2016-02-11",
  "dataLimiteInscricao":"2016-03-11"
}
```

- Response 200 (application/json)

```
{
  "data": [
    "CronogramaEditado":
    {
      "DataLimiteInscricao":"2016-11-09",
      "DataLimiteSubmissao":"2016-10-09",
      "DataLimiteRevisao":"2016-11-05",
      "IdEvento":"1"
    }
  ],
  "message":"Cronograma buscado com sucesso!"
}
```

- Response 400 (application/json)

```
{
  "message":"Erro ao editar Cronograma! Desculpe o transtorno,
  trabalharemos para identificar tal erro."
}
```

- Response 404 (application/json)

```
{
  "message":"Id de evento não existe!"
}
```

- Response 500 (application/json)

```
{
  "message": "Mensagem de erro proveniente do banco de dados"
}
```

Seção: Data de inscrição do cronograma de um evento  
[events/{idEvent}/schedule/date-register]

Deleta a data de inscrição do cronograma de um evento [DELETE]

- Parameters

- idEvent : 1 (number) - Um identificador único do Evento

- Response 200 (application/json)

```
{
  "data": [
    "CronogramaAtual":
    {
      "DataLimiteInscricao":null,
      "DataLimiteSubmissao":"2016-10-09",
      "DataLimiteRevisao":
      "2016-11-05",
      "IdEvento":"1"
    }
  ],
  "message":"Data Limite para inscrição excluída com sucesso!"
}
```

- Response 400 (application/json)  

```
{"message":"Erro ao deletar data limite para inscrição! Desculpe o transtorno, trabalharemos para identificar tal erro."}
```
- Response 404 (application/json)  

```
{"message":"Id de evento não existe!"}
```
- Response 500 (application/json)  

```
{"message": "Mensagem de erro proveniente do banco de dados" }
```

Seção: Data de submissão do cronograma de um evento  
[events/{idEvent}/schedule/date-submission]

Deleta a data de submissão do cronograma de um evento [DELETE]

- Parameters
  - idEvent : 1 (number) - Um identificador único do Evento
- Response 200 (application/json)  

```
{"data":[
  "CronogramaAtual":
  {"DataLimiteInscricao":null,"DataLimiteSubmissao":null,"DataLimiteRevisao":"2016-11-05","IdEvento":"1"} ],
  "message":"Data Limite para submissão excluída com sucesso!"}
```
- Response 400 (application/json)  

```
{"message":"Erro ao deletar data limite para submissão! Desculpe o transtorno, trabalharemos para identificar tal erro."}
```
- Response 404 (application/json)  

```
{"message":"Id de evento não existe!"}
```
- Response 500 (application/json)  

```
{"message": "Mensagem de erro proveniente do banco de dados" }
```

Seção: Data de submissão do cronograma de um evento  
[events/{idEvent}/schedule/date-revision]

Deleta a data de revisão do cronograma de um evento [DELETE]

- Parameters
  - idEvent : 1 (number) - Um identificador único do Evento
- Response 200 (application/json)
 

```
{
  "data": [
    {
      "CronogramaAtual": {
        "DataLimiteInscricao": null,
        "DataLimiteSubmissao": null,
        "DataLimiteRevisao": null,
        "IdEvento": "1"
      }
    }
  ],
  "message": "Data Limite para revisão excluída com sucesso!"
}
```
- Response 400 (application/json)
 

```
{
  "message": "Erro ao deletar data limite para revisão! Desculpe o transtorno, trabalharemos para identificar tal erro."
}
```
- Response 404 (application/json)
 

```
{
  "message": "Id de evento não existe!"
}
```
- Response 500 (application/json)
 

```
{
  "message": "Mensagem de erro proveniente do banco de dados"
}
```

## Group Committee

Seção: Membro ao comitê de um evento [events/{idEvent}/committee]

Adiciona nova relação entre evento, tipo de comitê e usuário [POST]

- Papéis com permissão
  - Administrador de Evento
- Parameters
  - idEvent : 1 (number) - Um identificador único do Evento
- Request (application/json)
 

```
{
  "emailMembro": "teste1",
  "tipoComite": "Organizacional"
}
```

- Response 200 (application/json)
 

```
{
  "data": [
    {
      "EmailMembro": "email1",
      "TipoComite": "Organizacional",
      "IdEvento": "1"
    }
  ],
  "message": "Usuário 'teste1' adicionado ao comitê 'Organizacional' do evento!"
}
```
- Response 400 (application/json)
 

```
{
  "message": "Erro ao adicionar membro ao comitê! Desculpe o transtorno, trabalharemos para identificar tal erro."
}
```

OR

```
{
  "message": "O usuário já é membro deste comitê!"
}
```

OR

```
{
  "message": "O tipo do comitê está incorreto! Deve ser 'Cientifico' ou 'Organizacional'"
}
```
- Response 404 (application/json)
 

```
{
  "message": "Id de evento não existe!"
}
```

OR

```
{
  "message": "O email de usuário informado não existe no sistema"
}
```
- Response 500 (application/json)
 

```
{
  "message": "Mensagem de erro proveniente do banco de dados"
}
```

Remove relação entre evento, tipo de comitê e usuário [DELETE]

- Papéis com permissão
  - Administrador de Evento
- Parameters
  - idEvent : 1 (number) - Um identificador único do Evento
- Request (application/json)
 

```
{
  "emailMembro": "teste1",
```



```
"tipoComite":"Organizacional"
}
```

- Response 200 (application/json)

```
{"data":[
  "EmailMembroRemovido":"email1",
  "TipoComite":"Organizacional",
  "message":"Usuário removido do comitê do evento!"}
```

- Response 400 (application/json)

```
{"message":"Erro ao remover membro do comitê! Desculpe o transtorno,
trabalharemos para identificar tal erro."}
```

OR

```
{"message":"O usuário não é membro deste comitê!"}
```

OR

```
{"message":"O tipo do comitê está incorreto! Deve ser "Cientifico" ou
"Organizacional" }
```

- Response 404 (application/json)

```
{"message":"Id de evento não existe!"}
```

OR

```
{"message":"O email de usuário informado não existe no sistema"}
```

- Response 500 (application/json)

```
{"message": "Mensagem de erro proveniente do banco de dados" }
```

## Group States

Seção: busca todos os Estados [states]

Retorna a lista dos Estados [GET]

- Response 200 (application/json)

```
{
```

```

"data": [
  {
    "Id": "1",
    "Nome": "Acre",
    "UF": "AC"
  }],
  "message": "Estados buscados com sucesso!"
}

```

- Response 400 (application/json)

```

{
  "message": "Erro ao buscar Estados! Desculpe o transtorno, trabalharemos para identificar tal erro."
}

```

- Response 500 (application/json)

```

{
  "message": "Mensagem de erro proveniente do banco de dados"
}

```

Seção: busca um estado específico [states/{idState}]

Retorna um estado[GET]

Mostra um usuário específico, através de seu identificador.

- Parameters
  - idState : 1 (number) - Um identificador único do Estado

- Response 200(application/json)

```

{
  "data": [
    {
      "Id": "1",
      "Nome": "Acre",
      "UF": "AC"
    }],
  "message": "Estado buscado com sucesso!"
}

```

- Response 400 (application/json)

```

{
  "message": "Erro ao buscar Estado! Desculpe o transtorno, trabalharemos para identificar tal erro."
}

```

- Response 404 (application/json)

```
{“message”:”Id de estado não existe!”}
```

- Response 500 (application/json)

```
{"message": "Mensagem de erro proveniente do banco de dados" }
```

## Group Cities

Seção: busca todas as Cidades [cities]

Retorna a lista das Cidades [GET]

- Response 200 (application/json)

```
{
  "data": [
    {
      "Id": "1",
      "Nome": "Afonso Cláudio",
      "idEstado": "8"
    },
    {
      "Id": "2",
      "Nome": "Água Doce do Norte",
      "idEstado": "8"
    },
    {
      "Id": "5563",
      "Nome": "Wanderlândia",
      "idEstado": "27"
    },
    {
      "Id": "5564",
      "Nome": "Xambioá",
      "idEstado": "27"
    }
  ],
  "message": "Cidades buscadas com sucesso!"
}
```

- Response 400 (application/json)

```
message": “Erro ao buscar Cidades! Desculpe o transtorno, trabalharemos para identificar tal erro.”}
```

- Response 500 (application/json)

```
{"message": "Mensagem de erro proveniente do banco de dados" }
```

Seção: busca uma cidade específica [cities/{idCity}]

Retorna um estado[GET]

- Parameters
  - idCity : 1 (number) - Um identificador único da Cidade

- Response 200(application/json)

```
{
  "data": [
    {
      "Id": "1",
      "Nome": "Afonso Cláudio",
      "idEstado": "8"
    },
  ],
  "message": "Cidade buscada com sucesso!"
}
```

- Response 400 (application/json)

```
message": "Erro ao buscar Cidade! Desculpe o transtorno, trabalharemos para identificar tal erro."}
```

- Response 404 (application/json)

```
{"message": "Id de estado não existe!"}
```

- Response 500 (application/json)

```
{"message": "Mensagem de erro proveniente do banco de dados" }
```

Seção: busca cidades de um estado específico [states/{idStates}/cities]

Retorna uma lista de cidades [GET]

- Parameters
  - idStates : 1 (number) - Um identificador único do Estado

- Response 200(application/json)

```
{
  "data": [
    {
      "Id": "203",
      "Nome": "Amapá",
      "idEstado": "4"
    },
    {
      "Id": "204",
      "Nome": "Calçoene",
      "idEstado": "4"
    },
    {
      "Id": "205",
      "Nome": "Cutias",
      "idEstado": "4"
    }
  ],
  "message": "Cidades buscadas com sucesso!"
}
```

- Response 400 (application/json)

```
{ "message": "Erro ao buscar Cidades! Desculpe o transtorno, trabalharemos para identificar tal erro." }
```

- Response 404 (application/json)

```
{ "message": "Id de estado não existe!" }
```

- Response 500 (application/json)

```
{ "message": "Mensagem de erro proveniente do banco de dados" }
```

## Group Addresses

Seção: busca todos os Endereços [addresses]

Retorna a lista dos Endereços [GET]

- Response 200 (application/json)

```
"data": [
```

```
{
  "Id": "1",
  "Cep": "12345678",
  "Logradouro": "EndereçoTeste1",
  "Complemento": "EndereçoTeste1",
  "IdCidade": "1"
},
{
  "Id": "2",
  "Cep": "87654321",
  "Logradouro": "EndereçoTeste2",
  "Complemento": "EndereçoTeste2",
  "IdCidade": "2"
}
],
"message": "Endereços buscados com sucesso!"
}
```

- Response 400 (application/json)

```
{ "message": "Erro ao buscar Endereços! Desculpe o transtorno, trabalharemos para identificar tal erro." }
```

- Response 500 (application/json)

```
{ "message": "Mensagem de erro proveniente do banco de dados" }
```

Seção: busca um endereço específico [addresses/{idAddress}]

Retorna um endereço[GET]

Mostra um endereço específico, através de seu identificador.

- Parameters
  - idAddress : 1 (number) - Um identificador único de Endereço

- Response 200(application/json)

```
"data": [
  {
    "Id": "1",
    "Cep": "12345678",
    "Logradouro": "EndereçoTeste1",
    "Complemento": "EndereçoTeste1",
    "IdCidade": "1"
  }
],
```

```
"Endereços buscado com sucesso!"  
}
```

- Response 404 (application/json)

```
{"message": "Id de estado não existe!"}
```

- Response 400 (application/json)

```
{"message": "Erro ao buscar Endereços! Desculpe o transtorno, trabalharemos para  
identificar tal erro."}
```

- Response 500 (application/json)

```
{"message": "Mensagem de erro proveniente do banco de dados" }
```

# Diagrama de Classes

