

Documentação REST ActFirst

Recurso Ações (/api/acoes)

Obter todos os tipos de ações existentes na base de dados
Vai retornar os tipos de ações existentes na base de dados
/api/acoes/tipoacoes
Parâmetros:
Sucesso (200): [{"tipoAcao_id":1,"nome":"Recolha de alimentos"}, {"tipoAcao_id":2,"nome":"Ajuda na organização"}, {"tipoAcao_id":3,"nome":"Limpar Jardim"}]
Erro: 500: Erro do servidor
Exemplo: <pre>let tipoAcoes = await \$.ajax({ url: "/api/acoes/tipoacoes", method: "get", dataType: "json" });</pre>

Vai buscar as ações que um voluntario se encontra a participar (Presente)
Vai buscar todas as ações que um utilizador se encontra a participar (Presente)
/api/acoes/participacao?userID=:ID(GET)
Parâmetros: user_id (obrigatório): ID do utilizador
Sucesso (200): [{"acao_id":2,"NomeOrganizacao":"Cruz Vermelha Portuguesa","organizacao_id":12,"localizacao":"Lisboa","lat":38.728245,"lng":-9.139271,"tipoAcao":2,"extraInfo":"Levar roupa adequada","email":"geral@CVP.pt","diaAcaoInicio":"15-12-20","horaAcaoInicio":"13:00","diaAcaoFim":"30-12-20","horaAcaoFim":"15:00","nome":"Ajuda na organização","numeroInscritos":1,"maximoPessoas":30}]
Erro: 500: Erro do servidor
Exemplo: <pre>let acoes = await \$.ajax({ url: "/api/acoes/participacao?userID="+utilizadorID, method: "get", dataType: "json" });</pre>

Inserir uma nova ação na base de dados

Vai criar uma ação

/api/acoes/ (POST)

Dados:

```
{
  "organizacao_id": 4, "latitude": 38.7237927, "longitude": 38.7237927, "localizacao": "Lisboa", "tipoAcao": 1, "extraInfo": "Levar roupa adequada", "diaAcaoInicio": "2020-12-12 15:00:00", "diaAcaoFim": "2020-12-30 16:30:00", "maximoPessoas": 20
}
```

Sucesso (200):

```
{
  "fieldCount": 0,
  "affectedRows": 1,
  "insertId": 14,
  "serverStatus": 2,
  "warningCount": 0,
  "message": "",
  "protocol41": true,
  "changedRows": 0
}
```

Erro:

500: Erro do servidor

Exemplo:

```
let data = {
  organizacao_id: utilizadorID,
  latitude: infoLocalizacao.position.lat,
  longitude: infoLocalizacao.position.lng,
  localizacao: infoLocalizacao.address.city,
  tipoAcao: acaoTipo,
  extraInfo: maisInfo,
  diaAcaoInicio: dataInicio + " " + horaInicio,
  diaAcaoFim: dataFim + " " + horaFim,
  maximoPessoas: maxPessoas,
}

let result = $.ajax({
  url: "/api/acoes",
  method: "post",
  data: JSON.stringify(data),
  contentType: "application/json",
  dataType: "json"
});

alert("Ação criada com sucesso!");
```

```

    }
    else {
        alert("Localização inválida!")
    }

});

}
catch(err) {
    console.log(err);
}
}
else {
    alert("Preencha os campos!");
}
}
}

```

Apagar uma ação criada(organização)

Remove todas as pessoas que estão a participar na ação e por sua vez remove a ação

/api/acoes?acao_id=:id&organizacao_id=: id, (delete)

Parâmetros:

pos (obrigatório): id da ação

pos (obrigatório): id da organização

Sucesso (200):

```

{
  "fieldCount": 0,
  "affectedRows": 1,
  "insertId": 0,
  "serverStatus": 2,
  "warningCount": 0,
  "message": "",
  "protocol41": true,
  "changedRows": 0
}

```

Erro:

500: Erro do servidor

Exemplo:

```
let acao = await $.ajax({
  url: "/api/acoes?acao_id="+acao_id+"&organizacao_id="+utilizadorID,
  method: "delete",
  dataType: "json"
});
```

Vai adicionar uma ação ao voluntario

Adiciona uma ação ao voluntario

/api/acoes/addacao

Parâmetros:

```
{
  "acao_id": 4,
  "user_id": 13
}
```

Sucesso (200):

```
{
  "fieldCount": 0,
  "affectedRows": 1,
  "insertId": 16,
  "serverStatus": 2,
  "warningCount": 0,
  "message": "",
  "protocol41": true,
  "changedRows": 0
}
```

Erro:

500: Erro do servidor

Exemplo:

```
let data = {
  acao_id: id,
  user_id: utilizadorID
}

try {

  let result = await $.ajax({
    url: "/api/acoes/addacao",
    method: "post",
    data: JSON.stringify(data),
    contentType: "application/json",
    dataType: "json"
  });
```

Vai buscar as ações que o utilizador ainda não se inscreveu
Vai filtrar as ações que existem na base de dados conforme a localização e a data
/api/acoes?userID=:id&localizacao=:text&userID=:data
Parâmetros: pos user_id (obrigatório): ID do utilizador pos localizacao (obrigatório): localização escolhida pelo utilizador pos data (obrigatório): Data escolhida pelo utilizador
Sucesso (200): [{"acao_id":3,"NomeOrganizacao":"Florinhas do Vouga","organizacao_id":11,"localizacao":"Lisboa","lat":38.718334,"lng":-9.150734,"tipoAcao":3,"nome":"Limpar Jardim","extraInfo":"Levar roupa adequada","email":"geral@FV.pt","diaAcaoInicio":"11-02-21","horaAcaoInicio":"14:00","diaAcaoFim":"12-02-21","horaAcaoFim":"19:00","maximoPessoas":80,"numeroInscritos":3}]
Erro: 500: Erro do servidor
Exemplo: <pre>let acoes = await \$.ajax({ url: "/api/acoes?userID="+utilizadorID+"&localizacao="+localizacao+"&data="+data, method: "get", dataType: "json" });</pre>

Vai buscar as ações que um voluntario já participou(passado)
Vai buscar todas as ações que um utilizador participou no passado(passado)
/api/acoes/futuras?userID=:ID(GET)
Parâmetros: pos user_id (obrigatório): ID do utilizador
Sucesso (200): [{"acao_id":3,"NomeOrganizacao":"Florinhas do Vouga","organizacao_id":11,"localizacao":"Lisboa","lat":38.718334,"lng":-9.150734,"tipoAcao":3,"extraInfo":"Levar roupa adequada","email":"geral@FV.pt","diaAcaoInicio":"11-02-21","horaAcaoInicio":"14:00","diaAcaoFim":"12-02-21","horaAcaoFim":"19:00","nome":"Limpar Jardim","numeroInscritos":4,"maximoPessoas":80}]
Erro: 500: Erro do servidor
Exemplo:

```
let acoes = await $.ajax({
  url: "/api/acoes/participadas?userID="+utilizadorID,
  method: "get",
  dataType: "json"
});
```

Vai buscar as ações que um voluntario irá participar (futuro)

Vai buscar todas as ações que um utilizador irá participar(futuro)

/api/acoes/futuras?userID=:ID(GET)

Parâmetros:

pos user_id (obrigatório): ID do utilizador

Sucesso (200):

```
[{"acao_id":4,"NomeOrganizacao":"Abrigo
Seguro","organizacao_id":4,"localizacao":"Lisboa","lat":38.723793,"lng":38.723793,"tipoAcao":1,"extra
Info":"Levar roupa adequada","email":"geral@AS.pt","diaAcaoInicio":"14-04-
21","horaAcaoInicio":"15:00","diaAcaoFim":"06-05-21","horaAcaoFim":"17:56","nome":"Recolha de
alimentos","numeroInscritos":1,"maximoPessoas":20}]
```

Erro:

500: Erro do servidor

Exemplo:

```
let acoes = await $.ajax({
  url: "/api/acoes/futuras?userID="+utilizadorID,
  method: "get",
  dataType: "json"
});
```

Recurso Utilizadores (/api/utilizadores)

Vai buscar o utilizador
Se o utilizador existir vai buscar a informação
/api/utilizadores/:pos (GET)
Parâmetros:
pos (obrigatório): ID do utilizador
Sucesso (200): { "user_id":2, "username":"Jorge", "email":"jorge@gmail.com", "organizacao":0, "moderador":1 }
Erro:
500: Erro do servidor
404: Eventos não foram encontrados
{ "msg": "This username does not exist!" }
Exemplo:
<pre>let user = await \$.ajax({ url: "/api/utilizadores/"+utilizadorID, method: "get", dataType: "json" });</pre>

Verificação de utilizador(login)
Vai verificar se o utilizador existe se o utilizador existir vai buscar a informação
/api/utilizadores?email=:text / (GET)
Parâmetros:
Email (obrigatório): Email do utilizador pretendido.
Sucesso (200): { "user_id":2, "username":"Jorge", "email":"jorge@gmail.com", "organizacao":0, "moderador":1 }
Erro:
500: Erro do servidor
Exemplo:
<pre>let data = { username: username, email: email } try { let utilizador = await \$.ajax({ url: "/api/utilizadores", method: "post", data: JSON.stringify(data), contentType: "application/json", dataType: "json"</pre>

```
});
```

Verificação de utilizador(login)

Vai verificar se o utilizador existe se o utilizador existir vai buscar a informação

/api/utilizadores?email=:text / (GET)

Parâmetros:

Email (obrigatório): Email do utilizador pretendido.

Sucesso (200):

```
{"user_id":2,"username":"Jorge","email":"jorge@gmail.com","organizacao":0,"moderador":1}
```

Erro:

500: Erro do servidor

Exemplo:

```
let utilizador = await $.ajax({  
  url: "/api/utilizadores?email="+email,  
  method: "get",  
  dataType: "json"  
});
```