

## Serviço REST

(2021/2022)



Hugo Duarte, 20190843  
Frederico Ramalho, 50034822

**Title:** getAllBuoys**Descrição:** Devolve as boias marítimas.**URL:** /api/buoys**Method:** GET**Success Response:****Code:** 200**Content:** [{"boia\_id":1,"boia\_nome":"River Gurara Popa","boia\_long":"38.4117",  
"boia\_lat":"-9.2166","boia\_profundidade":23,"boia\_poluicao":"Bom",  
"regiao\_id\_fk":1,"regiao\_id":1,"regiao\_nome":"Lisboa",  
"regiao\_globalid":1111026...}**Sample Call:**

```
let buoys = await $.ajax({  
  url: `/api/buoys`,  
  method: "get",  
  dataType: "json"  
});
```

**Title:** getBuoybyId**Descrição:** Devolve a boia marítima pelo seu id.**URL:** /api/buoys/:buoyId**Method:** GET**URL Params:****Required:**

buoyId = [integer]

example: buoyId: 2

**Success Response:****Code:** 200**Content:** {"boia\_id":1,"boia\_nome":"River Gurara  
Popa","boia\_long":"38.4117","boia\_lat":"-  
9.2166","boia\_profundidade":23,"boia\_poluicao":"Bom","regiao\_id\_fk":1,  
"regiao\_id":1,"regiao\_nome":"Lisboa","regiao\_globalid":1111026}**Error Response:****Code:** 404**Content:**

```
{msg: "Bóia não encontrada"}
```

**Sample Call:**

```
let buoy = await $.ajax({  
  url: `/api/buoys/${buoyId}`,  
  method: "get",  
  dataType: "json"  
});
```

**Title:** getAllHarbors**Descrição:** Devolve os portos marítimos.**URL:** /api/harbors**Method:** GET**Success Response:****Code:** 200**Content:** [{"porto\_id":1,"porto\_nome":"Marina de Sesimbra","porto\_long":"38.43603","porto\_lat":"-9.11542","regiao\_id\_fk":1,"regiao\_id":1,"regiao\_nome":"Lisboa","regiao\_globalid":1111026}...]**Sample Call:**

```
let harbors = await $.ajax({
  url: `/api/harbors`,
  method: "get",
  dataType: "json"
});
```

**Title:** getHarborById**Descrição:** Devolve o porto marítimo pelo seu id.**URL:** /api/buoys/:harborId**Method:** GET**URL Params:****Required:**

harborId = [integer]

example: harborId: 2

**Success Response:****Code:** 200**Content:** [{"porto\_id":1,"porto\_nome":"Marina de Sesimbra","porto\_long":"38.43603","porto\_lat":"-9.11542","regiao\_id\_fk":1,"regiao\_id":1,"regiao\_nome":"Lisboa","regiao\_globalid":1111026}]**Error Response:****Code:** 404**Content:**

```
{msg: "Porto não encontrado"}
```

**Sample Call:**

```
let harbor = await $.ajax({
  url: `/api/harbors/${harborId}`,
  method: "get",
  dataType: "json"
});
```

**Title:** getAllSeasons**Descrição:** Devolve as temporadas da aplicação.**URL:** /api/seasons**Method:** GET**Success Response:****Code:** 200**Content:** [{"temporada\_id":1,"temporada\_numero":1,"temporada\_datainicio":"2021-09-30T23:00:00.000Z","temporada\_datafim":"2021-10-30T23:00:00.000Z"}, {"temporada\_id":2,"temporada\_numero":2,"temporada\_datainicio":"2021-11-01T00:00:00.000Z","temporada\_datafim":"2022-01-31T00:00:00.000Z"}]**Sample Call:**

```
let seasons = await $.ajax({
  url: `/api/seasons`,
  method: "get",
  dataType: "json"
});
```

**Title:** getSeasonByCURRENT\_DATE**Descrição:** Devolve a temporada correspondente ao dia atual.**URL:** /api/seasons/curdate**Method:** GET**Success Response:****Code:** 200**Content:** {"temporada\_id":2,"temporada\_numero":2,"temporada\_datainicio":"2021-11-01T00:00:00.000Z","temporada\_datafim":"2022-01-31T00:00:00.000Z"}**Error Response:****Code:** 404**Content:**

```
{msg: "Season não encontrada"}
```

**Sample Call:**

```
let season = await $.ajax({
  url: `/api/seasons/curdate`,
  method: "get",
  dataType: "json"
});
```

<b>Title:</b> getUserById <b>Descrição:</b> Devolve o utilizador pelo seu id.
<b>URL:</b> /api/users/:userId
<b>Method:</b> GET
<b>URL Params:</b> <b>Required:</b> userId = [integer] example: userId: 2
<b>Success Response:</b> <b>Code:</b> 200 <b>Content:</b> { "util_id": 1, "util_nome": "Frederico", "util_password": "1234", "util_admin": false, "licenca_id_fk": 1, "licenca_id": 1, "licenca_nome": "Open Water", "licenca_profundidade": 15 }
<b>Error Response:</b> <b>Code:</b> 404 <b>Content:</b> <code>{ "msg": "Utilizador não encontrado" }</code>
<b>Sample Call:</b> <pre>let user = await \$.ajax({     url: `/api/users/\${userId}`,     method: "get",     dataType: "json" });</pre>

<b>Title:</b> getUserPointsBySeason <b>Descrição:</b> Devolve os pontos relativamente a uma temporada do utilizador.
<b>URL:</b> /api/users/:userId/points/season/:seasonId
<b>Method:</b> GET
<b>URL Params:</b> <b>Required:</b> userId = [integer] example: userId: 2 seasonId = [integer] example: seasonId: 2
<b>Success Response:</b> <b>Code:</b> 200 <b>Content:</b> { "sum": "20" }

**Error Response:****Code:** 404**Content:**

```
{msg: "Pontos não encontrados"}
```

**Sample Call:**

```
let points = await $.ajax({  
    url: `/api/users/${userId}/points/season/${seasonId}`,  
    method: "get",  
    dataType: "json"  
});
```

**Title:** getUserParticipations**Descrição:** Devolve as participações do utilizador por filtros.**URL:** /api/users/:userId/participations/:filteroption/season/:seasonId:**Method:** GET**URL Params:****Required:**

userId = [integer]

example: userId: 2

seasonId = [integer]

example: seasonId: 2

filteroption = [string]

example: filteroption: all | past | future

**Success Response:****Code:** 200**Content:**

```
[{"participacao_id":8,"participacao_validar":false,"participacao_pontos":0,  
"utilizador_id_fk":1,"evento_id_fk":10,"evento_id":10,"evento_data":"2021-11-  
27T00:00:00.000Z","evento_vagas":10,"boia_id_fk":2,"porto_id_fk":1,  
"temporada_id_fk":2,"boia_id":2,"boia_nome":"River Gurara  
Proa","boia_long":"38.4113","boia_lat":"-  
9.2177","boia_profundidade":28,"boia_poluicao":"Bom","regiao_id_fk":1}]...
```

**Sample Call:**

```
let participations = await $.ajax({  
    url:  
    `/api/users/${userId}/participations/${filteroption}/season/${seasonId}`,  
    method: "get",  
    dataType: "json"  
});
```

<b>Title:</b> login <b>Descrição:</b> Sistema de login.
<b>URL:</b> /api/users/login
<b>Method:</b> POST
<b>Data Params:</b> <b>Example:</b> <pre>"nome": [string], "password": [alphanumeric] } "nome": "Hugo", "password": "1234" }</pre>
<b>Error Response:</b> <b>Code:</b> 401 <b>Content:</b> <pre>{msg: "Nome ou password errada"}</pre>
<b>Sample Call:</b> <pre>let user = await \$.ajax({   url: '/api/users/login',   method: 'post',   dataType: 'json',   data: JSON.stringify(obj),   contentType: 'application/json' });</pre>

<b>Title:</b> insertEvent <b>Descrição:</b> Inserir um evento na aplicação.
<b>URL:</b> /api/events
<b>Method:</b> POST
<b>Data Params:</b> <b>Example:</b> <pre>"date": [date], "vagas": [int], "buoyId": [int], "harborId": [int] } "date": "2021-12-06" "vagas": 5, "buoyId": 3, "harborId": 2 }</pre>

**Error Response:****Code:** 400**Content:**

```
{err: "Número de vagas inválidas"}
```

**Code:** 400**Content:**

```
{err: "Evento já existe"}
```

**OR****Code:** 400**Content:**

```
{err: "Data inválida"}
```

**Sample Call:**

```
let result = $.ajax({
  url: `/api/events`,
  method: 'post',
  data: JSON.stringify(addEvent),
  contentType: 'application/json',
  error: function() {
    alert(arguments[0].responseJSON.err);
  },
  success: function(){
    getlasteventId();
    alert("Evento criado com sucesso!");
  }
});
```

**Title:** getAllAvailableEvents**Descrição:** Devolve todos os eventos disponíveis da aplicação.**URL:** /api/events/available**Method:** GET**Success Response:****Code:** 200**Content:**

```
[{"evento_id":9,"evento_data":"2021-12-26T00:00:00.000Z","evento_vagas":5,"boia_id_fk":1,"porto_id_fk":1,"temporada_id_fk":2,"boia_id":1,"boia_nome":"River Gurara Popa","boia_long":"38.4117","boia_lat":"-9.2166","boia_profundidade":23,"boia_poluicao":"Bom","regiao_id_fk":1,"porto_id":1,"porto_nome":"Marina de Sesimbra","porto_long":"38.43603","porto_lat":"-9.11542"}]
```

**Sample Call:**

```
let events = await $.ajax({
  url: `/api/events/available`,
  method: "get",
  dataType: "json"
});
```



**Title:** getParticipantsByEventId**Descrição:** Devolve todos os participantes de um evento.**URL:** /api/events/:eventId/participants**Method:** GET**URL Params:****Required:**

eventId = [integer]

example: eventId: 2

**Success Response:****Code:** 200**Content:** [{"participacao\_id":1,"util\_id":7,"util\_nome":"Mamadu",  
"util\_admin":true,"licenca\_id":3,"licenca\_nome":"Deep",  
"licenca\_profundidade":40},  
{ "participacao\_id":8,"util\_id":1,"util\_nome":"Frederico",  
"util\_admin":false,"licenca\_id":1,"licenca\_nome":"OpenWater",  
"licenca\_profundidade":15}...**Sample Call:**

```
let participants = await $.ajax({  
    url: `/api/events/${eventId}/participants`,  
    method: "get",  
    dataType: "json"  
});
```

**Title:** getAvailableEventsCountByBuoyId**Descrição:** Verifica se uma boia marítima está a ser ocupada por outro evento.**URL:** /api/events/available/buoy/:buoyId**Method:** GET**URL Params:****Required:**

buoyId = [integer]

example: buoyId: 2

**Success Response:****Code:** 200**Content:** {"count":"1"}**Error Response:****Code:** 400**Content:**

```
{msg: "Evento não encontrado"}
```

**Sample Call:**

```
let event = await $.ajax({  
    url: `/api/events/available/buoy/${buoyId}`,  
    method: "get",  
    dataType: "json"  
});
```

**Title:** getLastEventId**Descrição:** Devolve o id do último evento inserido.**URL:** /api/events/last**Method:** GET**Success Response:****Code:** 200**Content:** {"max":20}**Error Response:****Code:** 404**Content:**

```
{msg: "Evento não encontrado"}
```

**Sample Call:**

```
let event = await $.ajax({  
    url: `/api/events/last`,  
    method: "get",  
    dataType: "json"  
});
```

**Title:** RemoveEventById**Descrição:** Apaga o evento da base de dados pelo seu id.**URL:** /api/events/:eventId/delete**Method:** DELETE**URL Params:****Example:**

eventId = [integer]

example: eventId: 2

**Error Response:****Code:** 400**Content:**

```
{err: "Não pode cancelar um dia antes do evento!"}
```

**Sample Call:**

```
let result = $.ajax({
  url: `/api/events/${eventId}/delete`,
  method: 'DELETE',
  success: function(){
    alert("Evento removido com sucesso!");
  }
});
```

**Title:** insertParticipation**Descrição:** Inserir uma participação num evento da aplicação.**URL:** /api/participations**Method:** POST**Data Params:****Example:**

```
"eventId": [int],
"userId": [int],
}
"eventId": 3,
"userId": 5,
}
```

**Error Response:****Code:** 400**Content:**

```
{err: "Não há vagas disponíveis"}
```

**Code:** 400**Content:**

```
{err: "Já está inscrito no evento"}
```

**OR****Code:** 400**Content:**

```
{err: "Não tem os requisitos de certificação"}
```

**Sample Call:**

```
let result = $.ajax({
  url: `/api/participations`,
  method: 'post',
  data: JSON.stringify(addParticipation),
  contentType: 'application/json',
  error: function() {
    alert(arguments[0].responseJSON.err);
  },
  success: function(){
    alert("Ingressou no evento com sucesso!");
  }
});
```

**Title:** RemoveParticipationById**Descrição:** Apaga uma participação de um evento da base de dados.**URL:** /api/events/:participationId/delete**Method:** DELETE**URL Params:****Example:**

participationId = [integer]

example: participationId: 2

**Error Response:****Code:** 400**Content:**

```
{err: "Não pode cancelar um dia antes do evento!"}
```

**Sample Call:**

```
let result = $.ajax({
  url: `/api/participations/${participacaoId}/delete`,
  method: 'DELETE',
  success: function(){
    alert("Participação removida com sucesso!");
  },
  error: function() {
    alert(arguments[0].responseJSON.err);
  }
});
```