Documentação REST do projeto SportsPlus

Recurso Utilizadores (/api/users)

|  |
| --- |
| **Obter a informação de um utilizador**  Vai devolver a informação de um certo utilizador |
| /api/users?user\_nickname=:text&user\_id=:id **(get)** |
| Parâmetros:  **user\_nickname** (obrigatório): nickname do utilizador pretendido.  ou  **user\_id** (obrigatório): id do utilizador pretendido. |
| Sucesso (200):  {"user\_id":1,"user\_name":"Miguel","user\_nickname":"miguel","user\_gender":"M","user\_birthday":"1999-09-29T23:00:00.000Z","user\_email":"miguel@gmail.com"} |
| Erro:  500: Erro do servidor  404: Utilizador não foi encontrado  {msg: "User not found!"} |
| Exemplo:  let user = await $.ajax({                  url: "/api/users?user\_nickname="+usernameInput,                  method: "get",                  dataType: "json"              }); |

|  |
| --- |
| **Obter a informação dos eventos a participar**  Vai devolver a informação dos eventos que um certo utilizador está a participar |
| /api/users/:pos/attend **(get)**  Parâmetros:  **pos** (obrigatório): o id do utilizador |
| Sucesso (200):  [  {"event\_id":15,"event\_name":"Padel","event\_date":"19-02-2021 12:00","event\_min":1,"event\_max":3,"event\_private":1,"sport\_name":"Padel","players":{"totalPlayers":1,"list":[]}},  …] |
| Erro:  500: Erro do servidor  404: Eventos não foram encontrados  {msg: "Events not found!"} |
| Exemplo:  let events = await $.ajax({              url: "/api/users/"+user.user\_id+"/attend",              method: "get",              dataType: "json"          }); |

|  |
| --- |
| **Obter a informação dos eventos**  Vai devolver a informação dos eventos que um certo utilizador criou |
| /api/users/:pos/events **(get)** |
| Parâmetros:  **pos** (obrigatório): o id do utilizador |
| Sucesso (200):  [  {"event\_id":15,"event\_name":"Padel","event\_date":"19-02-2021 12:00","event\_min":1,"event\_max":3,"event\_private":1,"sport\_name":"Padel","players":{"totalPlayers":1,"list":[]}},  …] |
| Erro:  500: Erro do servidor  404: Eventos não foram encontrados  {msg: "Events not found!"} |
| Exemplo:  let data = {              nickname: nick,              event\_id: id          }          try {                let result = await $.ajax({                  url: "/api/events/adduserwhitelist",                  method: "post",                  data: JSON.stringify(data),                  contentType: "application/json",                  dataType: "json"              });              alert("User successfully added!");              window.location = "myevents.html";              } catch(err) {              console.log(err);              if (err.status == 404) {                  alert(err.responseJSON.msg);              }          } |

|  |
| --- |
| **Obter a informação dos eventos participados**  Vai devolver a informação dos eventos que um certo utilizador participou |
| /api/users/:pos/participated **(get)** |
| Parâmetros:  **pos** (obrigatório): o id do utilizador |
| Sucesso (200):  [  {"event\_id":15,"event\_name":"Padel","event\_date":"19-02-2021 12:00","event\_min":1,"event\_max":3,"event\_private":1,"sport\_name":"Padel","players":{"totalPlayers":1,"list":[]}},  …] |
| Erro:  500: Erro do servidor  404: Eventos não foram encontrados  {msg: "Events not found!"} |
| Exemplo:  let events = await $.ajax({              url: "/api/users/"+user.user\_id+"/participated",              method: "get",              dataType: "json"          }); |

|  |
| --- |
| **Inserir um novo utilizador**  Cria um novo utilizador |
| /api/users/ **(post)** |
| Dados:  { "user\_name": "Miguel Amaro", "user\_nickname": "miguel", "user\_gender": "M", "user\_birthday": "2020-09-30", "user\_email": "miguel@gmail.com" } |
| Sucesso (200):  {      "fieldCount": 0,      "affectedRows": 1,      "insertId": 25,      "serverStatus": 2,      "warningCount": 0,      "message": "",      "protocol41": **true**,      "changedRows": 0  } |
| Erro:  500: Erro do servidor  404: Eventos não foram encontrados  {msg: "Username already exists!"} |
| Exemplo:  let data = {              user\_name: name,              user\_nickname: nickname,              user\_gender: gender,              user\_birthday: birthday,              user\_email: email          }            try {                let result = await $.ajax({                  url: "/api/users",                  method: "post",                  data: JSON.stringify(data),                  contentType: "application/json",                  dataType: "json"              });              res.innerHTML = "User successfully registered!";              window.location = "login.html";            } catch(err) {              console.log(err);              if (err.status == 404) {                  //Vai mostrar uma mensagem se o utilizador ja existir                  res.innerHTML = err.responseJSON.msg;              }              if (err.responseJSON) {                  res.innerHTML = err.responseJSON.msg;              } else {                  res.innerHTML = "Could not create user!";              }          } |

|  |
| --- |
| **Adicionar utilizador como participante**  Adicione um certo utilizador como participante de um certo evento |
| /api/users/ **(post)** |
| Dados:  { "user\_id": 2, "event\_id": 3 } |
| Sucesso (200):  {      "fieldCount": 0,      "affectedRows": 1,      "insertId": 18,      "serverStatus": 2,      "warningCount": 0,      "message": "",      "protocol41": **true**,      "changedRows": 0  } |
| Erro:  500: Erro do servidor  404: Utilizador já a participar no evento  {msg: "You are already participating in the event!"}  404: Utilizador não estiver na whitelist do evento privado  {msg: "You are not on the whitelist of this event!"} |
| Exemplo:  let data = {              user\_id: user.user\_id,              event\_id: event.event\_id,              event\_private: event.event\_private          }            try {                let result = await $.ajax({                  url: "/api/users/attend",                  method: "post",                  data: JSON.stringify(data),                  contentType: "application/json",                  dataType: "json"              });                alert("A participar no evento com sucesso!");              closeModel(); //Fechar Model            } catch(err) {              console.log(err);              if (err.status == 404) {                  alert(err.responseJSON.msg);              }          } |

Recurso Desportos (/api/sports)

|  |
| --- |
| **Obter todos os desportos**  Obter todos os desportos/modalidades |
| /api/users/sports **(get)** |
| Sucesso (200):  [  {"id":1,"name":"Padel","image":"https://i.imgur.com/2Zu2q9k.png"},{"id":2,"name":"Ténis","image":"https://i.imgur.com/2Zu2q9k.png"},{"id":3,"name":"Futebol","image":"https://i.imgur.com/ajvXIcs.png"},{"id":4,"name":"Andebol","image":"https://i.imgur.com/2Iai3Ax.png"},{"id":5,"name":"Basketball","image":"https://i.imgur.com/nVLR3Jg.png"},{"id":6,"name":"Volleyball","image":"https://i.imgur.com/3Ln11NE.png"},{"id":7,"name":"Correr","image":"https://i.imgur.com/iLLITeL.png"},{"id":8,"name":"Caminhar","image":"https://i.imgur.com/S4ztS8b.png"},{"id":9,"name":"Bicicleta","image":"https://i.imgur.com/3W4y3vK.png"},{"id":10,"name":"Futsal","image":"https://i.imgur.com/Ho6ftRn.png"}  ] |
| Erro:  500: Erro do servidor  404: Desportos não foram encontrados  {msg: "Sports not found!"} |
| Exemplo:  let sports = await $.ajax({              url: "/api/sports",              method: "get",              dataType: "json"          }); |

Recurso Eventos (/api/events)

|  |
| --- |
| **Obter a informação dos eventos**  Vai devolver a informação de todos os eventos e o número de participantes que estão a participar no evento e uma lista de todos os jogadores a participar |
| /api/events?sportId =:id&clubId=:id&date=:text **(get)** |
| Parâmetros:  **sportId** (opcional): Id do desporto  **clubId** (opcional): Id do clube  **date** (opcional): date de participação do evento |
| Sucesso (200):  [  {"event\_id":5,"event\_name":"Football Game","event\_sport\_id":3,"event\_description":"Derbi :)","event\_date":"15-12-20 00:00","event\_local":"","event\_duration":90,"event\_max":22,"event\_min":10,"event\_private":0,"event\_creator\_id":2,"event\_creation\_date":"26-01-21 17:15","event\_club\_id":4,"sport\_name":"Futebol","sport\_image":"https://i.imgur.com/ajvXIcs.png","club\_name":"Rackets Pro - Saldanha","club\_local":"38.731314#-9.145042","players":{"totalPlayers":1,"list":[{"user\_id":5,"user\_name":"Rafael","user\_nickname":"rafael","user\_gender":"M","user\_birthday":"1997-09-29T23:00:00.000Z","user\_email":"Rafael@gmail.com","participant\_id":17,"participant\_user\_id":1,"participant\_event\_id":5}]}}  ] |
| Erro:  500: Erro do servidor  404: Eventos não encontrados  {msg: "Events not found!"} |
| Exemplo:  let events = await $.ajax({              url: "/api/events?sportId="+sportID+"&clubId="+clubID+"&date="+date,              method: "get",              dataType: "json"          }); |

|  |
| --- |
| **Obter a informação de um evento**  Vai devolver a informação de um certo evento. |
| /api/events/:pos **(get)** |
| Parâmetros:  **pos** (obrigatório): o id do evento |
| Sucesso (200):  {"event\_id":2,"event\_name":"Correr","event\_sport\_id":7,"event\_description":"Correr","event\_date":"2021-01-30 00:00","event\_local":"38.7250514#-9.1329159","event\_duration":2,"event\_max":10,"event\_min":2,"event\_private":0,"event\_creator\_id":2,"event\_creation\_date":"26-01-21 17:15","event\_club\_id":null} |
| Erro:  500: Erro do servidor  404: Eventos não foram encontrados  {msg: "Events not found!"} |
| Exemplo:  let event = await $.ajax({              url: "/api/events/"+event\_id,              method: "get",              dataType: "json"          }); |

|  |
| --- |
| **Apagar um evento**  Vai apagar um certo evento e apagar os participantes contidos nele. |
| /api/events/:pos **(delete)** |
| Parâmetros:  **pos** (obrigatório): o id do evento |
| Sucesso (200):  1 |
| Erro:  500: Erro do servidor |
| Exemplo:  let event = await $.ajax({              url: "/api/events/"+event\_id,              method: "delete",              dataType: "json"          }); |

|  |
| --- |
| **Inserir um novo Evento**  Criar um novo evento e adiciona o criador do evento como participante. |
| /api/events/ **(post)** |
| Dados:  { "name": "Padel", "sport": 1, "desc": "Padel com amigos", "date": "2020-02-01", "location": " 38.717126039425644#-9.184291609466129", "duration": 1, "max": 4, "min": 2, "private": false, "creator\_id": 1, "club": 1  } |
| Sucesso (200):  {      "fieldCount": 0,      "affectedRows": 1,      "insertId": 19,      "serverStatus": 2,      "warningCount": 0,      "message": "",      "protocol41": **true**,      "changedRows": 0  } |
| Erro:  500: Erro do servidor |
| Exemplo:  let data = {          name: event\_name,          desc: event\_desc,          sport: event\_sport,          club: event\_club,          date: d,          private: event\_privacy,          min: event\_min,          max: event\_max,          duration: event\_duration,          location: event\_location,          creator\_id: user.user\_id      }      try {          let result = await $.ajax({              url: "/api/events",              method: "post",              data: JSON.stringify(data),              contentType: "application/json",              dataType: "json"          });          box.style.display = "block";          box.style.backgroundColor = "#a8e063";          box.innerHTML = "Successfully created event <i class='fas fa-check'></i>";        } catch(err) {          console.log(err);          if (err.responseJSON) {              box.style.display = "block";              box.style.backgroundColor = "#ED213A";              box.innerHTML = err.responseJSON.msg + " <i class='fas fa-exclamation-triangle'></i>";          } else {              box.style.display = "block";              box.style.backgroundColor = "#ED213A";              box.innerHTML = "It was not possible to create the event <i class='fas fa-exclamation-triangle'></i>";          }      } |

|  |
| --- |
| **Inserir um novo utilizador na whitelist**  Insere um certo utilizador a uma whitelist de um certo evento. |
| /api/events/adduserwhitelist **(post)** |
| Dados:  { "nickname": "miguel", "event\_id": 1} |
| Sucesso (200):  {      "fieldCount": 0,      "affectedRows": 1,      "insertId": 19,      "serverStatus": 2,      "warningCount": 0,      "message": "",      "protocol41": **true**,      "changedRows": 0  } |
| Erro:  500: Erro do servidor  404: Utilizador já está na whitelist  {msg: "This user is already on the whitelist!"}  404: Utilizador não existe  {msg: "This user does not exist!"} |
| Exemplo:  let data = {              nickname: nick,              event\_id: id          }          try {                let result = await $.ajax({                  url: "/api/events/adduserwhitelist",                  method: "post",                  data: JSON.stringify(data),                  contentType: "application/json",                  dataType: "json"              });              alert("User successfully added!");              window.location = "myevents.html";              } catch(err) {              console.log(err);              if (err.status == 404) {                  alert(err.responseJSON.msg);              }          } |

|  |
| --- |
| **Editar informações de um evento**  Editar informações de um certo evento. |
| /api/events/ **(put)** |
| Dados:  { "event\_id": 1, "event\_name": "padel2",  "event\_desc": "blablabla" , "event\_date": "2020-02-02" , "event\_time": "18:30" , "event\_private": false, "event\_min": 2, "event\_max": 4, "event\_duration": 2} |
| Sucesso (200):  {      "fieldCount": 0,      "affectedRows": 0,      "insertId": 0,      "serverStatus": 2,      "warningCount": 0,      "message": "(Rows matched: 0  Changed: 0  Warnings: 0",      "protocol41": **true**,      "changedRows": 0  } |
| Erro:  500: Erro do servidor. |
| Exemplo:  let data = {              event\_id: id,              event\_name: name,              event\_desc: desc,              event\_date: date,              event\_time: time,              event\_private: private,              event\_min: min,              event\_max: max,              event\_duration: duration          }            try {                let result = await $.ajax({                  url: "/api/events",                  method: "put",                  data: JSON.stringify(data),                  contentType: "application/json",                  dataType: "json"              });              alert("Event information updated successfully!");              window.location = "myevents.html";              } catch(err) {              console.log(err);          } |

Recurso Clubes (/api/clubs)

|  |
| --- |
| **Obter informação de todos os clubes**  Vai devolver a informação de todos os clubes |
| /api/clubs **(get)** |
| Sucesso (200):  [  {"id":1,"name":"VivePadel","local":"38.7552957#-9.1040299"},{"id":2,"name":"Padel Expo","local":"38.785970#-9.093277"},{"id":3,"name":"Padel Spot - Olaias","local":"38.741643#-9.122992"},{"id":4,"name":"Rackets Pro - Saldanha","local":"38.731314#-9.145042"}  ] |
| Erro:  500: Erro do servidor  404: Clubes não foram encontrados  {msg: "Clubs not found!"} |
| Exemplo:  let clubs = await $.ajax({              url: "/api/clubs",              method: "get",              dataType: "json"          }); |

|  |
| --- |
| **Obter a informação de um clube**  Vai devolver a informação de um certo clube |
| /api/clubs/:pos **(get)** |
| Parâmetros:  **pos** (obrigatório): o id do clube |
| Sucesso (200):  {"id":1,"name":"VivePadel","local":"38.7552957#-9.1040299"} |
| Erro:  500: Erro do servidor  404: Clube não foi encontrado  {msg: "Club not found!"} |
| Exemplo:  let club = await $.ajax({                  url: "/api/clubs/"+club\_id,                  method: "get",                  dataType: "json"              }); |