# API

A aplicação que é executada no servidor é constituída por sete modelos distintos, cada um com um papel específico na aplicação. São eles os modelos *user*, *challenges*, *events*, *fights*, *friendRequest*, *idols* e *record*.

Para cada modelo da aplicação será feita uma breve descrição e será listado o conjunto de métodos associados a ele. Para cada método será enunciado pedido de *HTTP* inerente ao mesmo, o seu URL (com os parâmetros que o constituem) uma breve descrição e a sua resposta em caso de sucesso e em caso de insucesso e caso seja relevante o corpo do pedido.

*User* – Modelo que representa um utilizador.

|  |  |  |
| --- | --- | --- |
| GET | /logout | Logout |
| GET | /users/search | Search |
| GET | /users/:userId/friendList | friendList |
| GET | /users/:userId | Show |
| GET | /users | All |
| GET | /users/addemail | Addemail |
| GET | /reset/:token | middleForgotPassword |
| POST | /register | Create |
| POST | /users/:userId | Update |
| POST | /users/:userId/changepassword | changePassword |
| POST | /users/:userId/changemail | changeEmail |
| POST | /users/rankings | Rankings |
| POST | /forgot | forgotPassword |
| POST | /reset/:token | finalizeForgotPassword |

|  |  |
| --- | --- |
| Logout | |
| HTTP Request | GET |
| URL | /logout |
| Descrição | Efectua o logout de um utilizador |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Logout bem sucedido | | Resposta | N/A | |

|  |  |
| --- | --- |
| Search | |
| HTTP Request | GET |
| URL | /users/search?name=’…’ |
| Parâmetros | Query: name – parâmetro de pesquisa |
| Descrição | Efectua uma pesquisa sobre utilizadores dado um parâmetro de pesquisa. O parâmetro é comparado com o nome e o email do utilizador. |
| Responses | |  |  | | --- | --- | | Código | 400 | | Condição | A pesquisa não retornou nenhum resultado. | | Resposta | “There are no users with that name or email” | | Código | 200 | | Condição | A pesquisa foi bem sucedida | | Resposta | O resultado da pesquisa em format JSON | |

|  |  |
| --- | --- |
| *FriendList* | |
| HTTP Request | GET |
| URL | /users/:userId/friendList |
| Parâmetros | :userId – Identificador de um utilizador. |
| Descrição | Obtém a lista de amigos de um dado utilizador |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A obtenção da lista de amigos foi bem sucedida | | Resposta | *Friendlist* do utilizador em formato JSON. | |

|  |  |
| --- | --- |
|  | Show |
| HTTP Request | GET |
| URL | /users/:userId |
| Parâmetros | :userId – Identificador de um utilizador. |
| Descrição | Obtém um utilizador. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A obtenção do utilizador foi bem sucedida | | Resposta | Obtém o utilizador em formato JSON | |

|  |  |
| --- | --- |
|  | All |
| HTTP Request | GET |
| URL | /users |
| Descrição | Obtém todos os utilizadores. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A obtenção dos utilizadores foi bem sucedida | | Resposta | Obtém os utilizadores em formato JSON | |

|  |  |
| --- | --- |
| Addemail | |
| HTTP Request | GET |
| URL | /users/addemail?name=’’ |
| Parâmetros | Query: name – email para enviar o convite |
| Descrição | Envia um email para o endereço indicado para convidar o dono do endereço para se registar e utilizar a aplicação |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | O envio do email foi bem sucedido. | | Resposta | ‘An e-mail has been sent to ‘ + Endereço de email submetido + ‘ inviting him to join UDO!’ | | Código | 400 | | Condição | O endereço submetido já se encontra registado na aplicação | | Resposta | ‘That email is already registered in UDO!’ | |

|  |  |
| --- | --- |
|  | middleForgotPassword |
| HTTP Request | GET |
| URL | /reset/:token |
| Parâmetros | :token – Token que representa a validade do pedido de alteração da password. |
| Descrição | Mostra um formulário que permitirá alterar a password do utilizador que iniciou o processo de mudança de password caso o token esteja dentro do prazo de validade. Mostra uma página de erro caso contrário. |
| Responses | N/A |
| Nota | Esta função apenas será executada fora da aplicação móvel. Utilizando um browser, mesmo que seja de um dispositivo móvel. |

|  |  |
| --- | --- |
|  | Create |
| HTTP Request | POST |
| URL | /register |
| Descrição | Efectua o registo de um utilizador. |
| Responses | |  |  | | --- | --- | | Código | 400 | | Condição | Caso alguns elementos do corpo do pedido não passem nas validações. | | Resposta | Lista com os erros | | Código | 400 | | Condição | O email utilizado no registo já se encontra registado na aplicação | | Resposta | ‘Email is already in use’ | | Código | 400 | | Condição | Erro na adição do utilizador | | Resposta | ‘Error adding user’ | | Código | 200 | | Condição | Registo bem sucedido | | Resposta | N/A | |

|  |  |
| --- | --- |
|  | Update |
| HTTP Request | POST |
| URL | /users/:userId |
| Descrição | Efectua a edição de um utilizador. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Edição bem sucedida | | Resposta | JSON com o utilizador já actualizado. | |

|  |  |
| --- | --- |
|  | changePassword |
| HTTP Request | POST |
| URL | /users/:userId/changePassword |
| Descrição | Altera a password do utilizador indicado. |
| Parâmetros | :userId – Identificador de um utilizador |
| Responses | |  |  | | --- | --- | | Código | 400 | | Condição | Caso a nova password indicada não cumpra o tamanho entre 8 e 20 caractéres. | | Resposta | ‘Password must be between 8-20 characters long’ | | Código | 200 | | Condição | A actualização seja bem sucedida | | Resposta | N/A | | Código | 401 | | Condição | Caso a password corrente não seja igual à submetida para efeitos de autenticação. | | Resposta | ‘Wrong current password’ | |

|  |  |
| --- | --- |
|  | changeEmail |
| HTTP Request | POST |
| URL | /users/:userId/changeEmail |
| Descrição | Altera o email do utilizador indicado. |
| Parâmetros | :userId – Identificador de um utilizador |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A actualização seja bem sucedida | | Resposta | N/A | | Código | 401 | | Condição | Caso a password ou email correntes não sejam iguais aos submetidos para efeitos de autenticação. | | Resposta | ‘Wrong email or password’ | |

|  |  |
| --- | --- |
|  | Rankings |
| HTTP Request | POST |
| URL | /users/rankings |
| Descrição | Obtém os rankings da aplicação e para um dado utilizador. |
| Responses | |  |  | | --- | --- | | Código | 400 | | Condição | Caso o utilizador em questão não exista. | | Resposta | ‘User does not exist! Something went wrong’ | | Código | 400 | | Condição | Caso a base de dados não contenha nenhum combate. | | Resposta | ‘There are no fights in the database’ | | Código | 200 | | Condição | Em caso de sucesso | | Resposta | JSON com todos os rankings. | |

|  |  |
| --- | --- |
|  | forgotPassword |
| HTTP Request | POST |
| URL | /forgot |
| Descrição | Cria um token e envia um email para alteração da password. |
| Responses | |  |  | | --- | --- | | Código | 400 | | Condição | Caso o email indicado não esteja registado. | | Resposta | “An account with such email doesn’t exist” | | Código | 200 | | Condição | Caso o email seja enviado com sucesso | | Resposta | ‘An e-mail has been sent to ‘ + email indicado + ‘ with further instructions’ | |
| Corpo do pedido | email: Email indicado para a alteração da password. |
| Nota | Esta função apenas será executada fora da aplicação móvel. Utilizando um browser, mesmo que seja de um dispositivo móvel. |

|  |  |
| --- | --- |
|  | finalizeForgotPassword |
| HTTP Request | POST |
| URL | /reset/:token |
| Descrição | Finaliza a alteração da password. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a alteração ocorra com sucesso. | | Resposta | 'Success! Your password has been changed, you can now return to the application and login with your new password.' | |
| Nota | Esta função apenas será executada fora da aplicação móvel. Utilizando um browser, mesmo que seja de um dispositivo móvel. |

Challenges – Modelo que representa um desafio que um utilizador pode fazer a um amigo.

|  |  |  |
| --- | --- | --- |
| GET | /challenges/all | listChallenges |
| GET | /challenges/:userId/count/pending | countPending |
| GET | /challenges/:userId/get/pending | getPending |
| GET | /challenges/:userId/get/ongoing | getOngoing |
| GET | /challenges/:userId/get/ended | getEnded |
| GET | /challenges/:userId/:challengeId/decline | declineChallenge |
| POST | /challenges/:userId/:challengeId/accept | acceptChallenge |
| POST | /challenges | createChallenge |

|  |  |
| --- | --- |
|  | listChallenges |
| HTTP Request | GET |
| URL | /challenges/all |
| Descrição | Obtém todos os desafios existentes na base de dados. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados seja bem sucedida. | | Resposta | JSON com todos os desafios na base de dados. | |

|  |  |
| --- | --- |
|  | countPending |
| HTTP Request | GET |
| URL | /challenges/:userId/count/pending |
| Parâmetros | :userId – Identificador de um utilizador. |
| Descrição | Conta o número de desafios pendentes que um utilizador tem. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a contagem dos dados seja bem sucedida. | | Resposta | JSON com o número de desafios pendentes | |

|  |  |
| --- | --- |
|  | getPending |
| HTTP Request | GET |
| URL | /challenges/:userId/get/pending |
| Parâmetros | :userId – Identificador de um utilizador. |
| Descrição | Obtém todos os desafios pendentes de um dado utilizador. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados seja bem sucedida. | | Resposta | JSON com todos os desafios pendentes de um utilizador. | |

|  |  |
| --- | --- |
|  | getOngoing |
| HTTP Request | GET |
| URL | /challenges/:userId/get/ongoing |
| Parâmetros | :userId – Identificador de um utilizador. |
| Descrição | Obtém todos os desafios de um dado utilizador que ainda não terminaram. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados seja bem sucedida. | | Resposta | JSON com todos os desafios de um utilizador que ainda estão a decorrer. | |

|  |  |
| --- | --- |
|  | getEnded |
| HTTP Request | GET |
| URL | /challenges/:userId/get/ended |
| Parâmetros | :userId – Identificador de um utilizador. |
| Descrição | Obtém todos os desafios de um dado utilizador que já terminaram. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados seja bem sucedida. | | Resposta | JSON com todos os desafios de um utilizador que já terminaram. | |

|  |  |
| --- | --- |
|  | declineChallenge |
| HTTP Request | GET |
| URL | /challenges/:userId/:challengeId/decline |
| Parâmetros | :userId – Identificador de um utilizador.  :challengeId – Identificador de um desafio. |
| Descrição | Rejeita um desafio |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados seja bem sucedida. | | Resposta | N/A | |

|  |  |
| --- | --- |
|  | acceptChallenge |
| HTTP Request | POST |
| URL | /challenges/:userId/:challengeId/accept |
| Parâmetros | :userId – Identificador de um utilizador.  :challengeId – Identificador de um desafio. |
| Descrição | Aceita um desafio |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados seja bem sucedida. | | Resposta | N/A | |

|  |  |
| --- | --- |
|  | createChallenge |
| HTTP Request | POST |
| URL | /challenges |
| Descrição | Cria um desafio |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados seja bem sucedida. | | Resposta | N/A | |

Events – Modelo que representa os eventos onde os combates se passaram.

|  |  |  |
| --- | --- | --- |
| GET | /events | showAllEvents |
| GET | /events/:eventId | showEvent |
| GET | /eventsbyuser/:userId | showEventsByUser |
| GET | /events/:eventId/remove | removeEvent |
| POST | /events/add | createEvent |
| POST | /events/:eventId/update | updateEvent |

|  |  |
| --- | --- |
|  | showAllEvents |
| HTTP Request | GET |
| URL | /events |
| Descrição | Obtém todos os eventos na base de dados. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados seja bem sucedida. | | Resposta | JSON com todos os eventos na base de dados. | |

|  |  |
| --- | --- |
|  | showEvent |
| HTTP Request | GET |
| URL | /events/:eventId |
| Parâmetros | :eventId – Identificador de um evento. |
| Descrição | Obtém um evento através do seu identificador. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados seja bem sucedida. | | Resposta | JSON com o evento pretendido. | |

|  |  |
| --- | --- |
|  | showEventsByUser |
| HTTP Request | GET |
| URL | /eventsbyuser/:userId |
| Parâmetros | :userId – Identificador de um utilizador. |
| Descrição | Obtém todos os eventos criados por um utilizador. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados seja bem sucedida. | | Resposta | JSON com todos os eventos de um determinado utilizador. | |

|  |  |
| --- | --- |
|  | showEventsByUser |
| HTTP Request | GET |
| URL | /eventsbyuser/:userId |
| Parâmetros | :userId – Identificador de um utilizador. |
| Descrição | Obtém todos os eventos criados por um utilizador. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados seja bem sucedida. | | Resposta | JSON com todos os eventos de um determinado utilizador. | |

|  |  |
| --- | --- |
|  | removeEvent |
| HTTP Request | GET |
| URL | /events/:eventId/remove |
| Parâmetros | :eventId – Identificador de um evento. |
| Descrição | Remove o evento com o identificador indicado. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso o evento seja removido com sucesso. | | Resposta | N/A | |

|  |  |
| --- | --- |
|  | addEvent |
| HTTP Request | POST |
| URL | /events/add |
| Descrição | Cria um evento. |
| Responses | |  |  | | --- | --- | | Código | 400 | | Condição | Caso exista um evento duplicado | | Resposta | N/A | | Código | **400** | | Condição | Caso existam campos sobre o evento que não foram preenchidos. | | Resposta | ‘Please fill all the required fields’ | | Código | 400 | | Condição | Existem para o mesmo utilizador dois eventos no mesmo dia | | Resposta | ‘You can not have two events on the same day’ | | Código | **400** | | Condição | Existe um evento do utilizador com o mesmo nome | | Resposta | ‘You already have an event with the same name’ | | Código | 200 | | Condição | Caso o evento seja criado com sucesso. | | Resposta | N/A | |

|  |  |
| --- | --- |
|  | updateEvent |
| HTTP Request | POST |
| URL | /events/:eventId/update |
| Descrição | Actualiza um evento |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso o evento seja actualizado com sucesso. | | Resposta | N/A | |

Fights – Modelo que define os combates de um utilizador. Estes combates podem ser entre utilizadores que sejam amigos, utilizadores contra outro judoca ou de ídolos do utilizador contra outros judocas.

|  |  |  |
| --- | --- | --- |
| GET | /fights/:userId | fightsByUser |
| GET | /fights | allFights |
| GET | /fights/:userId/mine | allUserParticipates |
| GET | /fights/:userId/statistics | retrieveStatistics |
| GET | /fights/:userId/opponents | listOpponents |
| GET | /fights/:userId/stats/byopponent | statsVersusOpponent |
| GET | /fights/:userId/stats/byevent | statsOnEvent |
| GET | /fights/:userId/stats/byfriend | statsVersusFriend |
| GET | /fights/:userId/stats/between | statsBetween |
| GET | /fights/remove/:fightId | removeFight |
| POST | /fights | createFight |
| POST | /fights/update/:fightId | updateFight |

|  |  |
| --- | --- |
|  | fightsByUser |
| HTTP Request | GET |
| URL | /fights/:userId |
| Parâmetros | :userId – Identificador de um utilizador |
| Descrição | Obtém todos os combates inseridos por um utilizador |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados ocorra com sucesso | | Resposta | JSON com todos os combates que um utilizador inseriu. | |

|  |  |
| --- | --- |
|  | allFights |
| HTTP Request | GET |
| URL | /fights |
| Descrição | Obtém todos os combates na base de dados |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados ocorra com sucesso | | Resposta | JSON com todos os combates na base de dados. | |

|  |  |
| --- | --- |
|  | allUserParticipates |
| HTTP Request | GET |
| URL | /fights/:userId/mine |
| Parâmetros | :userId – Identificador de um utilizador |
| Descrição | Obtém todos os combates na base de dados em que um determinado utilizador participa. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados ocorra com sucesso | | Resposta | JSON com os combates em que utilizador participa | |

|  |  |
| --- | --- |
|  | retrieveStatistics |
| HTTP Request | GET |
| URL | /fights/:userId/statistics |
| Parâmetros | :userId – Identificador de um utilizador |
| Descrição | Obtém as estatísticas de um utilizador. Calcula-as através de todos os combates que o utilizador participa. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados ocorra com sucesso | | Resposta | JSON com as estatísticas | |

|  |  |
| --- | --- |
|  | listOpponents |
| HTTP Request | GET |
| URL | /fights/:userId/opponents |
| Parâmetros | :userId – Identificador de um utilizador |
| Descrição | Devolve todos os oponentes de um determinado utilizador. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados ocorra com sucesso | | Resposta | JSON com todos os oponentes. | |

|  |  |
| --- | --- |
|  | statsVersusOpponent |
| HTTP Request | GET |
| URL | /fights/:userId/stats/byopponent |
| Parâmetros | :userId – Identificador de um utilizador  query – oponente – nome do oponente. |
| Descrição | Devolve as estatísticas de um utilizador contra um determinado oponente. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados ocorra com sucesso | | Resposta | JSON com as estatísticas contra um determinado oponente. | |

|  |  |
| --- | --- |
|  | statsOnEvent |
| HTTP Request | GET |
| URL | /fights/:userId/stats/byevent |
| Parâmetros | :userId – Identificador de um utilizador  query – event – identificador do evento. |
| Descrição | Devolve as estatísticas de um utilizador contra um determinado oponente. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados ocorra com sucesso | | Resposta | JSON com as estatísticas num determinado evento. | |

|  |  |
| --- | --- |
|  | statsBetween |
| HTTP Request | GET |
| URL | /fights/:userId/stats/between |
| Parâmetros | :userId – Identificador de um utilizador  query – begin – data de início query - end – data de fim |
| Descrição | Devolve as estatísticas de um utilizador entre duas datas. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção dos dados ocorra com sucesso | | Resposta | JSON com as estatísticas entre duas datas. | |

|  |  |
| --- | --- |
|  | removeFight |
| HTTP Request | GET |
| URL | /fights/remove/:fightId |
| Parâmetros | :fightId – Identificador de um combate |
| Descrição | Remove um combate |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a remoção ocorra com sucesso | | Resposta | N/A | |

|  |  |
| --- | --- |
|  | createFight |
| HTTP Request | POST |
| URL | /fights |
| Descrição | Cria um combate |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a criação ocorra com sucesso | | Resposta | N/A | |

|  |  |
| --- | --- |
|  | updateFight |
| HTTP Request | POST |
| URL | /fights/update/:fightId |
| Parâmetros | :fightId – Identificador de um combate |
| Descrição | Actualiza um combate |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a actualização ocorra com sucesso | | Resposta | N/A | |

FriendRequest – Modelo que define um pedido de amizade entre dois utilizadores.

|  |  |  |
| --- | --- | --- |
| GET | /friendRequests/:userId/all | getAllRequests |
| GET | /friendRequests/:userId/received | getReceivedRequests |
| GET | /friendRequests/:userId/count/received | countReceivedRequests |
| GET | /friendRequests/:userId/count/sent | countSentRequests |
| GET | /friendRequests/:requestId/accept | acceptRequest |
| GET | /friendRequests/:requestId/decline | declineRequest |
| POST | /friendRequests/:userId/addrequest | createRequest |

|  |  |
| --- | --- |
|  | getAllRequests |
| HTTP Request | GET |
| URL | /friendRequests/:userId/all |
| Parâmetros | :userId – Identificador de um utilizador |
| Descrição | Obtém todos os pedidos de amizade de um utilizador. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção ocorra com sucesso | | Resposta | JSON com todos os pedidos de amizade. | |

|  |  |
| --- | --- |
|  | getReceivedRequests |
| HTTP Request | GET |
| URL | /friendRequests/:userId/received |
| Parâmetros | :userId – Identificador de um utilizador |
| Descrição | Obtém os pedidos de amizade que um utilizador recebeu e ainda respondeu. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a obtenção ocorra com sucesso | | Resposta | JSON com todos os pedidos de amizade que recebeu e ainda estão pendentes. | |

|  |  |
| --- | --- |
|  | countReceivedRequests |
| HTTP Request | GET |
| URL | /friendRequests/:userId/count/received |
| Parâmetros | :userId – Identificador de um utilizador |
| Descrição | Obtém o número de pedidos de amizade que um utilizador recebeu e ainda respondeu. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a contagem ocorra com sucesso | | Resposta | JSON com o número de pedidos de amizade que o utilizador recebeu e ainda estão pendentes. | |

|  |  |
| --- | --- |
|  | countSentRequests |
| HTTP Request | GET |
| URL | /friendRequests/:userId/count/sent |
| Parâmetros | :userId – Identificador de um utilizador |
| Descrição | Obtém o número de pedidos de amizade que um utilizador enviou. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | Caso a contagem ocorra com sucesso | | Resposta | JSON com o número de pedidos de amizade que o utilizador enviou. | |

|  |  |
| --- | --- |
|  | acceptRequest |
| HTTP Request | GET |
| URL | /friendRequests/:requestId/accept |
| Parâmetros | :requestId – Identificador de um pedido |
| Descrição | Aceita um pedido de amizade. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A aceitação executou com sucesso. | | Resposta | N/A | |

|  |  |
| --- | --- |
|  | declineRequest |
| HTTP Request | GET |
| URL | /friendRequests/:requestId/decline |
| Parâmetros | :requestId – Identificador de um pedido |
| Descrição | Rejeita um pedido de amizade. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A rejeição executou com sucesso. | | Resposta | N/A | |

|  |  |
| --- | --- |
|  | createRequest |
| HTTP Request | POST |
| URL | /friendRequests/:userId/addrequest |
| Parâmetros | :userId – Identificador de um pedido |
| Corpo do pedido | to – Email do utilizador que se pretende adicionar  fromName – Nome do utilizador que enviou o pedido  fromEmail – Email do utilizador que enviou o pedido. |
| Descrição | Cria um pedido de amizade entre dois utilizadores. |
| Responses | |  |  | | --- | --- | | Código | 400 | | Condição | O utilizador tentou enviar um pedido de amizade a si próprio. | | Resposta | ‘You can't send friend requests to yourself!’ | | Código | 400 | | Condição | Caso o pedido seja para um utilizador que já tenha enviado um pedido igual. | | Resposta | ‘That user already sent you a request, check your notifications!’ | | Código | 400 | | Condição | Caso o pedido seja duplicado. | | Resposta | ‘You've already sent an invite to that user’ | | Código | 200 | | Condição | Caso o envio do pedido execute com sucesso. | | Resposta | JSON com o pedido | | Código | **400** | | Condição | O utilizador para o qual se tentou enviar o pedido não existe. | | Resposta | ‘That user doesn't exist’ | |

Idol – Modelo que define um ídolo. Cada utilizador tem os seus ídolos, assim a aplicação permite que cada utilizador tenha uma lista de ídolos e que para cada ídolo existam (ou não) combates realizados pelos mesmos. O objetivo é que um utilizador consiga comparar as suas estatísticas com as estatísticas do seu ídolo.

|  |  |  |
| --- | --- | --- |
| GET | /idols | getAllIdols |
| GET | /idols/:userId | getUserIdols |
| GET | /idols/:idolId/statistics | getIdolsStatistics |
| POST | /idols | createIdol |

|  |  |
| --- | --- |
|  | getAllIdols |
| HTTP Request | GET |
| URL | /idols |
| Descrição | Obtém todos os ídolos na base de dados |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A obtenção executou com sucesso. | | Resposta | JSON com todos os ídolos da base de dados | |

|  |  |
| --- | --- |
|  | getUserIdols |
| HTTP Request | GET |
| URL | /idols/:userId |
| Parâmetros | :userId – Identificador de um utilizador |
| Descrição | Obtém todos os ídolos de um determinado utilizador. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A obtenção executou com sucesso. | | Resposta | JSON com os ídolos de um determinado utilizador. | |

|  |  |
| --- | --- |
|  | getIdolStatistics |
| HTTP Request | GET |
| URL | /idols/:idolId/statistics |
| Parâmetros | :idolId – Identificador de um ídolo |
| Descrição | Obtém as estatísticas de um determinado ídolo. |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A obtenção executou com sucesso. | | Resposta | JSON com as estatísticas de um determinado ídolo. | |

|  |  |
| --- | --- |
|  | createIdol |
| HTTP Request | POST |
| URL | /idols |
| Descrição | Cria um ídolo |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A obtenção executou com sucesso. | | Resposta | JSON com as estatísticas de um determinado ídolo. | |

Records – Modelo que define momento na “história” do utilizador enquanto judoca. Tem o objetivo de funcionar de forma semelhante aos “*Life Events”* do *Facebook* mas aplicado ao Judo. Os *records* na aplicação podem ser inseridos manualmente ou aparecer automaticamente caso o sistema detecte que o utilizador melhorou uma estatística, ou mudou de grau no Judo.

|  |  |  |
| --- | --- | --- |
| GET | /recordsbyuser/:userId | getRecordsUser |
| POST | /records/add | createRecord |
| POST | /records/update | updateRecord |
| POST | /records/delete | deleteRecord |

|  |  |
| --- | --- |
|  | getRecordsUser |
| HTTP Request | GET |
| URL | /records/add |
| Descrição | Cria um record |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A obtenção executou com sucesso. | | Resposta | JSON com todos os records de um utilizador. | |

|  |  |
| --- | --- |
|  | createRecord |
| HTTP Request | POST |
| URL | /idols |
| Descrição | Cria um ídolo |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A criação ocorreu com sucesso. | | Resposta | N/A. | | Código | 400 | | Condição | Caso a inserção na base de dados resulte num erro. | | Resposta | ‘Error’ | | Código | 400 | | Condição | Caso já exista um record igual no mesmo dia. | | Resposta | ‘You can't have equal records on the same day.’ | |

|  |  |
| --- | --- |
|  | updateRecord |
| HTTP Request | POST |
| URL | /records/update |
| Descrição | Actualiza um record |
| Responses | |  |  | | --- | --- | | Código | 200 | | Condição | A actualização ocorreu com sucesso. | | Resposta | N/A | | Código | 400 | | Condição | O record não existe na base de dados. | | Resposta | ‘Such record doesn't exist’ | | Código | 400 | | Condição | Caso o record não seja editável. | | Resposta | ‘Error, you can't edit this record’ | | Código | 400 | | Condição | Caso a actualização resulte num record que é igual a outro que já exista no mesmo dia na base de dados. | | Resposta | ‘You can't have equal records on the same day’. | | Código | 400 | | Condição | Caso a inserção na base de dados result num erro. | | Resposta | ‘Error’ | |