***POST /api/account***

Permite a autenticação de um utilizador. Este *endpoint* recebe como *header* de *Authorization* o *ID Token* recebido do *OAuth* da Google.

O código *HTTP* de resposta poderá ser 200, caso o utilizador se tenha autenticado com sucesso ou 400 caso o utilizador não tenha uma conta de utilizador.

O corpo da resposta poderá conter o id único do utilizador, que será utilizado nos outros pedidos que a *API* disponibiliza, caso este tenha o código 200. Em caso de erro o corpo será composto por uma mensagem ilustrativa do erro.

**• *PUT /api/account***

Permite a criação de uma conta de utilizador. Este *endpoint* recebe como *header* de *Authorization* o *ID Token* recebido do *OAuth* da Google.

O código *HTTP* de resposta poderá ser 200, caso a criação da conta tenha tido sucesso, ou 409 caso já exista uma conta com o id obtido.

Como o *endpoint* anterior em caso de sucesso no corpo da resposta estará o id único do utilizador, ou uma mensagem descritiva do erro ocorrido.

**• *PUT /api/registerDevice***

Associa um dispositivo a conta de utilizador.

Este *endpoint* é composto por 4 parâmetros:

* *Authorization*: O *header authorization* deve conter o valor do *id* de utilizador que foi obtido quando se realizou a operação de *login*/criação da conta.
* *deviceIdentifier*: *ID* único que identifica um dispositivo. Caso se trate de um dispositivo móvel e pretenda utilizar o servido de GCM, este *id* terá que ser o *id* fornecido pela *API* da *Firebase*.
* *useSockets*: indica se este dispositivo pretende receber notificações através de *WebSockets* ou *GCM*. De notar que apenas dispositivos móveis poderão utilizar o serviço de *GCM*.
* *deviceName:* o nome do dispositivo.

**• *GET /api/socket***

*Endpoint* utilizado pelos dispositivos que pretendem obter informação através de *WebSockets*. A resposta deste *endpoint* contém no corpo o URI para o qual o dispositivo se deverá conectar.

Apesar de existir apenas um *URI* para o qual os dispositivos se poderão conectar, este *endpoint* poderá ser vantajoso caso se pretenda escalar o serviço pois apenas será necessário alterar uma componente (servidor), sem haver a necessidade de alterar as outras componentes.

**• *PUT /api/push***

Permite colocar conteúdo textual na base de dados.

Tem como parâmetros:

* *Authorization*: O *header authorization* deve conter o valor do *id* de utilizador que foi obtido quando se realizou a operação de *login*/criação da conta.
* data: Parâmetro que deverá conter o texto copiado.

**• *POST /api/push***

Permite colocar imagens na base de dados.

Tem como parâmetros:

* *Authorization*: O *header authorization* deve conter o valor do *id* de utilizador que foi obtido quando se realizou a operação de *login*/criação da conta.
* file: Parâmetro que deverá conter a imagem copiada.