FOLIO & FLEX – REST API documentation

Title: Get company accounts

Description: Gets companies from the 'utilizador' database table

URL: Approve/getUsers

Method: GET

Success response: Code 200

Content: {id: [int], email: [string], nome: [string], pass: [string], tipo_utilizador: [string], approved: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Approve company account

Description: Approves company account

URL: Approve/UpdateUser

Method: POST

Data Params: {userid: [string]}

Success response: Code 200

Content: 0

Error response: Code: 500

Content: {error: [error]}

Title: Reject company account

Description: Rejects company account

URL: Approve/RejectUser

Method: POST

Data Params: {userid: [string]}

Success response: Code 200

Content: 0

Error response: Code: 500

Content: {error: [error]}

Title: Deactivate company account

Description: Deactivates company account

URL: Approve/DeactivateUser

Method: POST

Data Params: {userid: [string]}

Success response: Code 200

Content: 0

Error response: Code: 500

Content: {error: [error]}

Title: Get logged user information

Description: Gets logged user’s information from the ‘utilizador’ database table

URL: Home/getloggedinUserType

Method: GET

Data Params: email: [string]

Success response: Code 200

Content: {id: [int], email: [string], nome: [string], pass: [string], tipo_utilizador: [string], approved: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Get logged professional user information

Description: Gets logged user’s information from the ‘profissional database table

URL: Home/getloggedinUserInformationProfissional

Method: GET

Data Params: email: [string]

Success response: Code 200

Content: {email: [string], nome: [string], data_nascimento: [date], genero: [string], headline: [string], descricao: [string], localidade: [string], visualizacao_empresas: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Get logged company user information

Description: Gets logged company’s information from the ‘empresa’ database table

URL: Home/getloggedinUserInformationEmpresa

Method: GET

Data Params: email: [string]

Success response: Code 200

Content: {email: [string], nome: [string], descricao: [string], site: [string]}

Error response: Code: 500

Content: {error: [error]}

Title: Get logged company user information

Description: Gets logged company’s information from the ‘empresa’ database table

URL: Home/getloggedinUserInformationEmpresa

Method: GET

Data Params: email: [string]

Success response: Code 200

Content: {email: [string], nome: [string], descricao: [string], site: [string]}

Error response: Code: 500

Content: {error: [error]}

Title: Get logged user ID

Description: Gets ID from logged user

URL: Home/getloggedinUserID

Method: GET

Data Params: email: [string]

Success response: Code 200

Content: {id: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Get list of professional users

Description: Gets all the professional users from the database table 'utilizador' excluding the logged in user

URL: Home/getUsersProfissionais

Method: GET

Data Params: id : [int]

Success response: Code 200

Content: {id: [int], email: [string], nome: [string], pass: [string], tipo_utilizador: [string], approved: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Get data from professional users

Description: Gets the data of the profissional users from the ‘profissional’ table excluding the logged user

URL: Home/getUsersProfissionaisInformation

Method: GET

Data Params: email : [string]

Success response: Code 200

Content: {email: [string], nome: [string], data_nascimento: [date], genero: [string], headline: [string], descricao: [string], localidade: [string], visualizacao_empresas: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Get professional users’ age

Description: Calculates the age of all users with a query to the ‘profissional’ database table

URL: Home/getUsersProfissionaisAge

Method: GET

Success response: Code 200

Content: {age: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Send friend request

Description: Sends friend request from a user to another

URL: Home/sendFriendRequest

Method: POST

Data Params: {id: [int], userid: [int]}

Success response: Code 200

Content: 0

Error response: Code: 500

Content: {error: [error]}

Title: Get friend requests

Description: Gets all friend requests from the database table ‘pedido_amizade’

URL: Home/getFriendRequests

Method: POST

Success response: Code 200

Content: {id_origem: [int], id_destino: [int], aprovado: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Get all approved companies and professional users

Description: Gets all approved companies and professional users from the database table 'utilizador'

URL: Home/getUsers

Method: GET

Success response: Code 200

Content: {id: [int], email: [string], nome: [string], pass: [string], tipo_utilizador: [string], approved: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Get friends

Description: Gets logged user's friends from the database table 'amigo'

URL: Home/getFriends

Method: GET

Data Params: email : [string]

Success response: Code 200

Content: {email_utilizador: [string], email_amigo: [string]}

Error response: Code: 500

Content: {error: [error]}

Title: Accept friend request

Description: Accepts friend request

URL: Home/acceptFriend

Method: POST

Data Params: {destino: [int], origem: [int], useremail: [string], destinoemail: [string]}

Success response: Code 200

Content: 0

Error response: Code: 500

Content: {error: [error]}

Title: Cancel friend request

Description: cancels friend request

URL: Home/cancelFriendRequest

Method: POST

Data Params: {userid: [int]}

Success response: Code 200

Content: 0

Error response: Code: 500

Content: {error: [error]}

Title: Deny friend request

Description: deletes received friend request

URL: Home/deleteFriendRequest

Method: POST

Data Params: {destino: [int], origem: [int]}

Success response: Code 200

Content: 0

Error response: Code: 500

Content: {error: [error]}

Title: Delete friend

Description: deletes friend from friends list

URL: Home/deleteFriend

Method: POST

Data Params: {destino: [int], origem: [int], emailUser: [string]}

Success response: Code 200

Content: 0

Error response: Code: 500

Content: {error: [error]}

Title: Get all users

Description: Gets all users from the database table 'utilizador'

URL: /getUsers

Method: GET

Success response: Code 200

Content: {id: [int], email: [string], nome: [string], pass: [string], tipo_utilizador: [string], approved: [int]}

Error response: Code: 500

Content: {error: "Internal Server error"}

Title: Professional user registration

Description: Registers user of type 'profissional'

URL: /registoP

Method: POST

Data Params: {nomeP: [string], emailP: [string], passP: [string], birthdayP: [date], genderP: [string]}

Success response: Code 201

Content: {id : [int]}

Error response: Code: 500

Content: {error: "Internal Server error"}

Title: Company user registration

Description: Registers user of type ‘empresa’

URL: /registoE

Method: POST

Data Params: {nomeE: [string], emailE: [string], passE: [string]}

Success response: Code 201

Content: {id : [int]}

Error response: Code: 500

Content: {error: “Internal Server error”}

Title: User login

Description: Login

URL: /login

Method: POST

Data Params: {emailLogin: [string], passLogin: [string]}

Success response: Code 200

Content: {result : [int]}

Error response: Code: 500

Content: {error: “Internal Server error”}

Title: Get profile user data

Description: Gets profile user data from database table ‘utilizador’

URL: Profile/getProfileType/:userid

Method: GET

URL Params: required: userid = [int]

Data Params: {email : [string]}

Success response: Code 200

Content: {id: [int], email: [string], nome: [string], pass: [string], tipo_utilizador: [string], approved: [int]}

Error response: Code: 500

Content: error

Title: Get professional profile user data

Description: Gets profile user data from database table ‘profissional’

URL: Profile/getProfileInformationProfissional/:userid

Method: GET

URL Params: required: userid = [int]

Data Params: {email : [string]}

Success response: Code: 200

Content: {email: [string], nome: [string], data_nascimento: [date], genero: [string], headline: [string], descricao: [string], localidade: [string], visualizacao_empresas: [int]}

Error response: Code: 500

Content: error

Title: Get company profile user data

Description: Gets profile user data from database table ‘empresa’

URL: Profile/getProfileInformationEmpresa/:userid

Method: GET

URL Params: required: userid = [int]

Data Params: {email : [string]}

Success response: Code: 200

Content: {email: [string], nome: [string], descricao: [string], site: [string]}

Error response: Code: 500

Content: error

Title: Get professional profile user birthdate

Description: Gets the user’s birthdate from the database table ‘profissional’ with the written format

URL: Profile/getProfileBirthdate/:userid

Method: GET

URL Params: required: userid = [int]

Data Params: {email : [string]}

Success response: Code: 200

Content: {data: [date]}

Error response: Code: 500

Content: error

Title: Get professional profile user experiences

Description: Gets the user's professional experiences from the database table 'experiencia_trabalho'

URL: Profile/getProfileExperiences/:userid

Method: GET

URL Params: required: userid = [int]

Data Params: {email : [string]}

Success response: Code: 200

Content: {id: [int], email_profissional: [string], nome_empresa: [string], descricao: [string], regime: [string], localizacao: [string], tipo_localizacao: [string], trabalho_atual: [int], data_inicio: [date], data_fim: [date], cargo: [string]}

Error response: Code: 500

Content: error

Title: Get professional profile user experiences' start and end dates

Description: Gets the user's professional experiences start and end dates from the database table 'experiencia_trabalho' with the written format

URL: Profile/getExperienceDates/:userid

Method: GET

URL Params: required: userid = [int]

Data Params: {email : [string]}

Success response: Code: 200

Content: {id: [int], datainicio: [date], datafim: [date]}

Error response: Code: 500

Content: {error: [error]}

Title: Get professional profile user educations

Description: Gets the user's educations from the database table 'educacao'

URL: Profile/getProfileEducations/:userid

Method: GET

URL Params: required: userid = [int]

Data Params: {email : [string]}

Success response: Code: 200

Content: {id: [int], email_profissional: [string], descricao: [string], atividades: [string], media: [decimal], atual: [int], data_inicio: [date], data_fim: [data], nome_curso: [string], tipo_curso: [string], estabelecimento_ensino: [string]}

Error response: Code: 500

Content: {error: [error]}

Title: Get professional profile user educations' start and end dates

Description: Gets the user's educations start and end dates from the database table 'educacao' with the written format

URL: Profile/getEducationDates/:userid

Method: GET

URL Params: required: userid = [int]

Data Params: {email : [string]}

Success response: Code: 200

Content: {id: [int], datainicio: [date], datafim: [date]}

Error response: Code: 500

Content: {error: [error]}

Title: Delete experience

Description: Deletes experience from profile

URL: Profile/deleteExperience/:userid

Method: POST

URL Params: required: userid = [int]

Data Params: {id: [int]}, {email: [string]}

Success response: Code: 200

Content: 0

Error response: Code: 500

Content: {error: [error]}

Title: Delete education

Description: Deletes education from profile

URL: Profile/deleteEducation/:userid

Method: POST

URL Params: required: userid = [int]

Data Params: {id: [int]},{email: [string]}

Success response: Code: 200

Content: 0

Error response: Code: 500

Content: {error: [error]}

Title: Edit professional user's introduction

Description: Edits professional user's introduction

URL: Profile/editIntro/:userid

Method: POST

URL Params: required: userid = [int]

Data Params: {nome: [string], birth: [date], genero: [string], headline: [string], localidade: [string], visualizacao: [int]}, {email: [string]}

Success response: Code: 200

Content: {id: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Edit company user's introduction

Description: Edits company user's introduction

URL: Profile/editIntroEmpresa/:userid

Method: POST

URL Params: required: userid = [int]

Data Params: {url: [string]}, {email: [string]}

Success response: Code: 200

Content: {id: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Edit professional user's description

Description: Edits professional user's description

URL: Profile/editDescription/:userid

Method: POST

URL Params: required: userid = [int]

Data Params: {description: [string]}, {email: [string]}

Success response: Code: 200

Content: {id: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Edit company user's description

Description: Edits company user's description

URL: Profile/editDescriptionEmpresa /:userid

Method: POST

URL Params: required: userid = [int]

Data Params: {description: [string]}, {email: [string]}

Success response: Code: 200

Content: {id: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Add experience

Description: Adds professional experience to user's profile

URL: Profile/addExperience/:userid

Method: POST

URL Params: required: userid = [int]

Data Params: {email_profissional: [string], cargo: [string], regime: [string], nome_empresa: [string], localizacao: [string], tipo_localizacao: [string], trabalho_atual: [int], data_inicio: [date], data_fim: [date], descricao: [string]}, {email: [string]}

Success response: Code: 201

Content: {id: [int]}

Error response: Code: 500

Content: {error: [error]}

Title: Add education

Description: Adds education to user's profile

URL: Profile/addEducation/:userid

Method: POST

URL Params: required: userid = [int]

Data Params: {email_profissional: [string], estabelecimento_ensino: [string], tipo_curso: [string], nome_curso: [string], atual: [int], data_inicio: [date], data_fim: [date], media: [decimal], atividades: [string], descricao: [string]},{email: [string]}

Success response: Code: 201

Content: {id: [int]}

Error response: Code: 500

Content: {error: [error]}