IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

Curso:

Engenharia Informática

Ano Letivo 2017/2018

Docente: João Ventura

Relatório Técnico - Computação Distribuída

Relatório realizado por:

-Tomás Santos 160221032

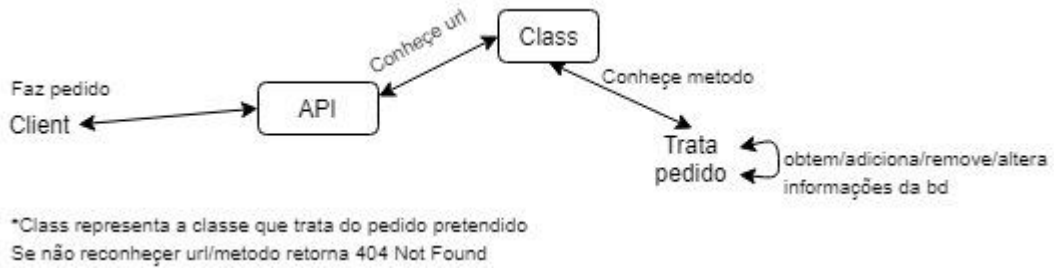
-Tiago Neto 160221086

-Hugo Ferreira 160221089

Turma 1

Funcionamento geral

Esta aplicação serve como gestor de projetos e suas determinadas tarefas, no qual um utilizador ao fazer a sua autenticação tem acesso aos seus projetos, podendo criar/alterar/eliminar os mesmos. Cada projeto tem um conjunto de tarefas podendo visualizar a informação de cada uma delas, bem como alterar/eliminar as mesmas, tendo também a possibilidade de criar mais tarefas. Um utilizador pode alterar as suas informações pessoais. Os utilizadores que não tem conta podem registar-se gratuitamente na aplicação, sendo que existirá um username único entre todos os utilizadores. Todo os projetos bem como as suas tarefas só estão disponíveis para os criadores dos mesmos, isto é, se um utilizador tentar aceder a um projeto que não é seu não terá permissão para o aceder.
Ao iniciar a API, esta fica à escuta de pedidos HTTP vindo de clientes. Os endpoints dos pedidos disponíveis são:
/login - Métodos GET-Aceder à página de login
/dashboard - Métodos GET - Aceder ao painel de informação de projetos e tarefas
/api/user/ - Métodos GET -Aceder a informação do utilizador autenticado e PUT - Alterar a informação do utilizador autenticado
/api/user/login/ - Métodos POST - Faz o login do utilizador
/api/user/logout/ - Métodos GET - Faz logout do utilizador
/api/user/register/ - Métodos POST - Faz o registo de um novo utilizador
/api/projects/ - Métodos GET - Acede aos projetos do utilizador autenticado e POST - Adiciona um novo projeto
/api/projects/<idProject>/ - Métodos GET - Acede à informação do projeto, PUT - Altera a informação do projeto e DELETE - Elimina o projeto
/api/projects/<idProject>/tasks/ - Métodos GET - Acede à lista de tarefas de um projeto e POST - Adiciona uma nova tarefa
/api/projects/<idProject>/tasks/<idTask>/ - Métodos GET - Acede à informação da tarefa, PUT - Altera a informação da tarefa e DELETE - Elimina a tarefa
/admin - Acesso à zona de administração



As mensagens entre servidor-cliente com os endpoints que comecem por /api são feitas através de json(application/json).

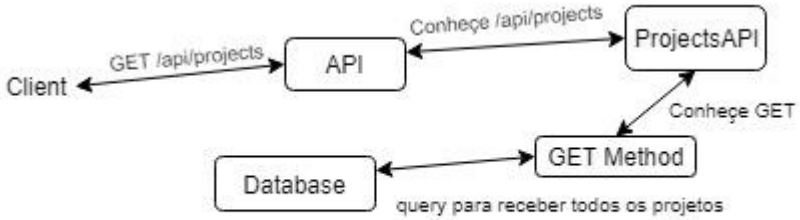
Funcionamento técnico

Classes/Módulos utilizados:
Main - Módulo que serve para a inicialização da app e configuração da mesma.
Models - Módulo que relativa aos modelos da base de dados, classes e métodos da mesma,
Views - Módulo que faz o roteamento dos diferentes pedidos vindos dos clientes e trata dos mesmos.
UrlsTester- Módulo de testes à api.
Validators - Classe que transforma um pedido http, em resposta http com a informação pretendida.
log.log - Representa a informação guardada sobre os pedidos efetuados.

Admin

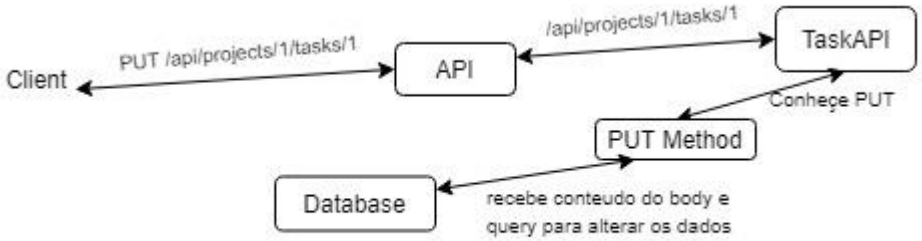
Para aceder a parte de administrador use o endpoint /admin, terá acesso a toda a informação relativa à base de dados

Exemplo de pedido para receber projetos do utilizador autenticado



* Neste caso o cliente iria rececer uma lista de objetos json(cada um corresponde a um projeto)
No caso de não conheçer o endpoint/método retorn 404 Not Found

Exemplo de pedido para atualizar uma tarefa



* Neste caso o cliente iria rececer um objeto json com {'result': 'Task atualizada com sucesso!'}
Caso o utilizador não tivesse aceso a a task pretendida recebia {'task': 'Task nao encontrada!'}

Por opção decidimos usar o http code 404, quando um utilizador tenta aceder a informações que não lhe pertencem por questões de segurança, pois neste caso ele nunca irá saber se o conteúdo a aceder não existe ou se não é permitido ele aceder. Caso se tivéssemos optado pelo 403 Forbidden ele iria ficar a saber que o conteúdo a aceder existia.

Instalações necessárias ao funcionamento

Para a api funcionar é necessário instalar os seguintes “componentes”:

- flask_sqlalchemy
- flask-admin
- flask_login
- flask_restful
- flask

Deployment na cloud

O projeto encontra-se na cloud segundo o link **http://hugofer.pythonanywhere.com/.**

Extras

1. Utilização de bootstrap
2. Sistema de Logs
3. Deployment na cloud(http://hugofer.pythonanywhere.com/)
4. Implementação de unittests relevantes
5. Utilização de ORMS, tais como:
    5.1. SQLAlchemy
6. Utilização de plugins, tais como:
    6.1. flask-admin
    6.2. flask-restful
    6.3. flask_login