Coding API REST

Introduction

Dans le cadre d’une Préparation Opérationnelle à l’Emploi (POE) de Pôle Emploi, nous avons suivi une formation d’ingénieur tests via IB-formation afin de potentiellement intégrer l’entreprise Harmonic.

Dans ce cadre, nous vous présentons notre projet final.

Sommaire

Introduction générale

Définition du projet

Réalisation

Démo

Tests

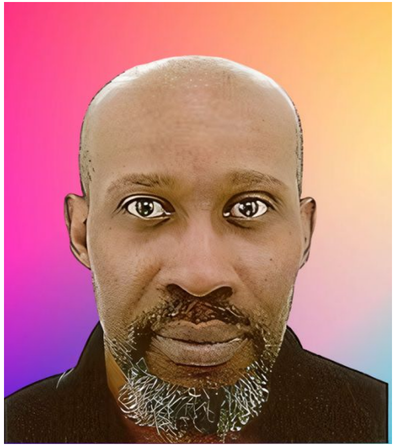
Présentation du sujet

Le sujet final est de développer une API REST

Cette API servira à tester de manière automatique tous codes qui lui seront soumis en langage Python et/ou langage C.

Elle devra être sécurisée.

Team



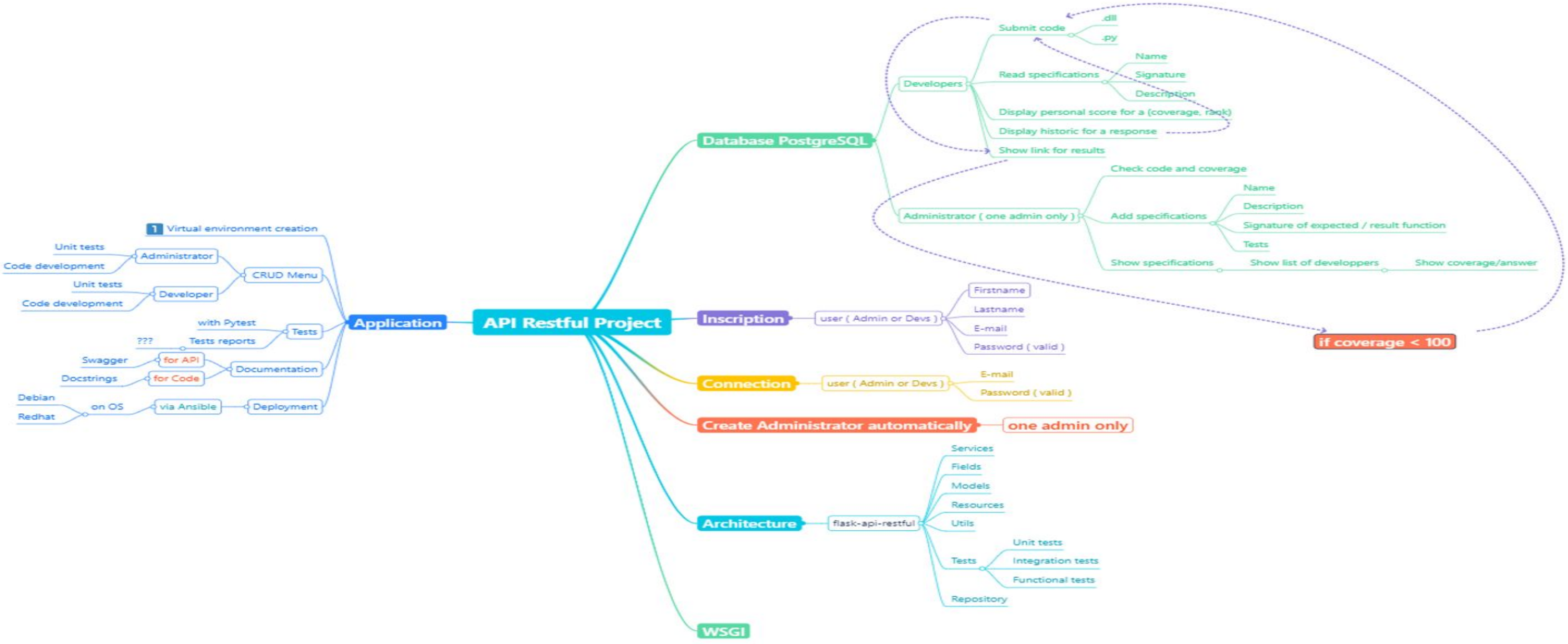
David DONISA

Ruddy MOUNIAMA MOUNICAN

Kévin BARDOUX

DÉFINITION DU PROJET

Notre réflexion sur le projet



Nos outils

PC PyCharm

Confluence

Jira Software

Bitbucket

git

BCrypt HASH

JWT

PostgreSQL

POSTMAN

pytest

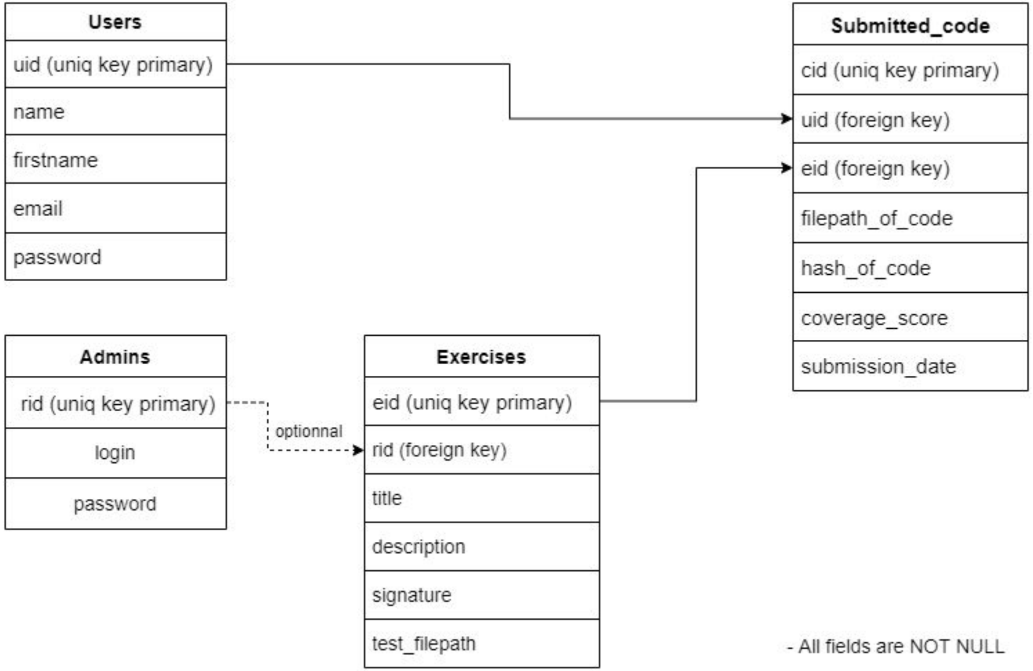
Flask

SQLAlchemy

Swagger

python

Définition de la Base de Données



Users
uid (uniq key primary)
name
firstname
email
password

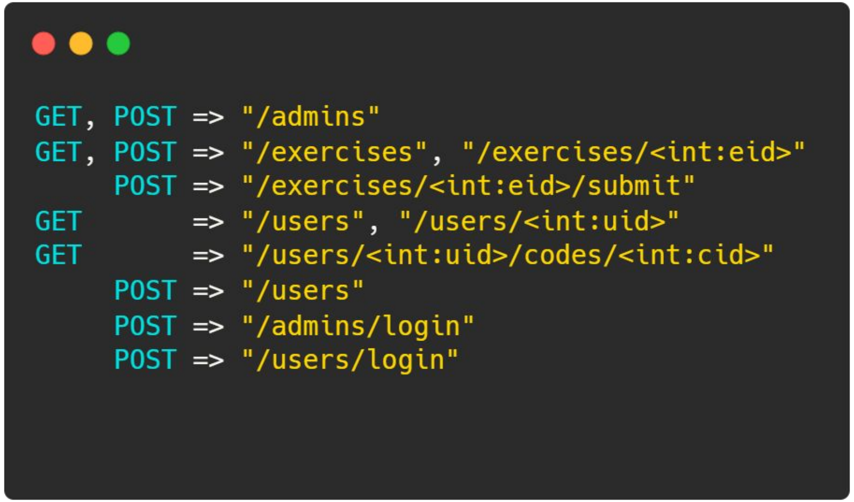
Admins
rid (uniq key primary)
login
password

Exercises
eid (uniq key primary)
rid (foreign key)
title
description
signature
test_filepath

Submitted_code
cid (uniq key primary)
uid (foreign key)
eid (foreign key)
filepath_of_code
hash_of_code
coverage_score
submission_date

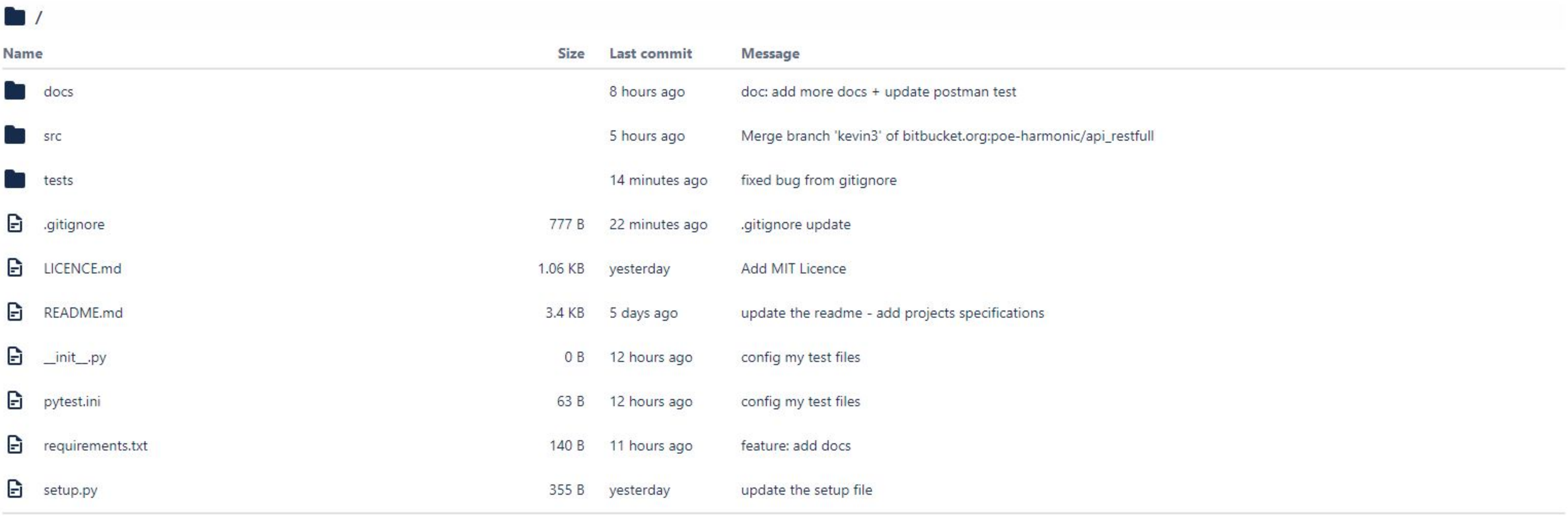
- All fields are NOT NULL

Les Routes



```
GET, POST => "/admins"
GET, POST => "/exercises", "/exercises/<int:eid>"
     POST => "/exercises/<int:eid>/submit"
GET       => "/users", "/users/<int:uid>"
GET       => "/users/<int:uid>/codes/<int:cid>"
     POST => "/users"
     POST => "/admins/login"
     POST => "/users/login"
```

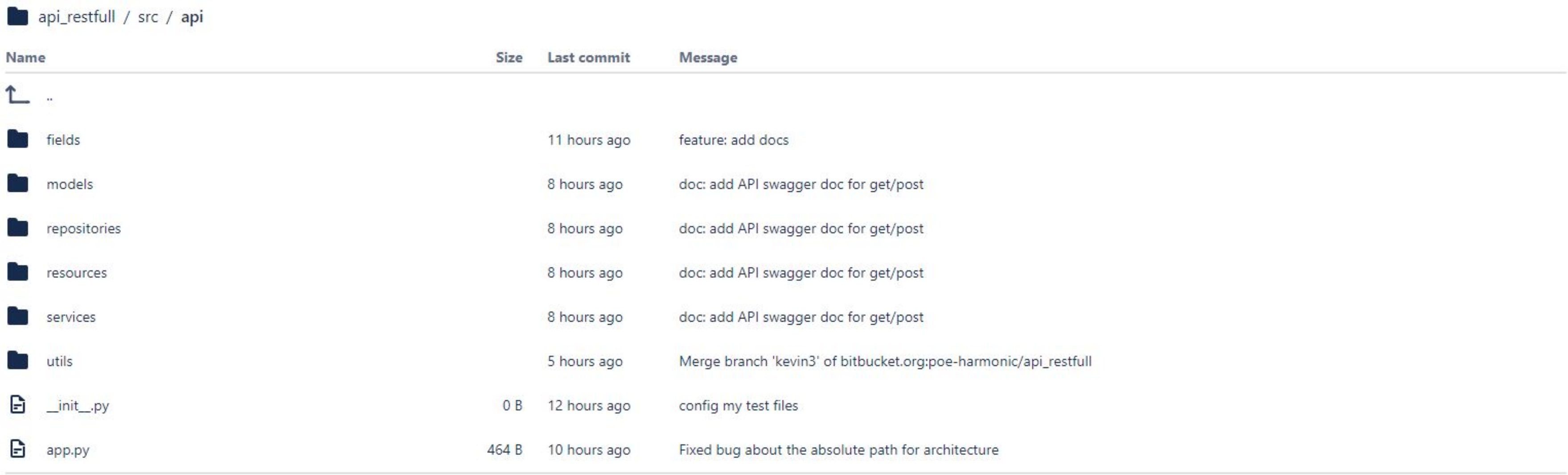
Structure globale



/

Name	Size	Last commit	Message
docs		8 hours ago	doc: add more docs + update postman test
src		5 hours ago	Merge branch 'kevin3' of bitbucket.org:poe-harmonic/api_restfull
tests		14 minutes ago	fixed bug from gitignore
.gitignore	777 B	22 minutes ago	.gitignore update
LICENCE.md	1.06 KB	yesterday	Add MIT Licence
README.md	3.4 KB	5 days ago	update the readme - add projects specifications
__init__.py	0 B	12 hours ago	config my test files
pytest.ini	63 B	12 hours ago	config my test files
requirements.txt	140 B	11 hours ago	feature: add docs
setup.py	355 B	yesterday	update the setup file

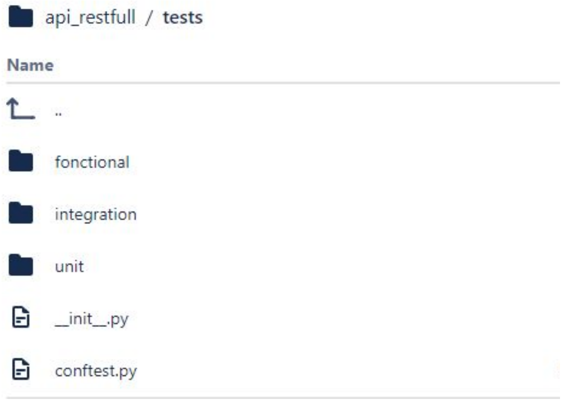
Structure API



api_restfull / src / api

Name	Size	Last commit	Message
..			
fields		11 hours ago	feature: add docs
models		8 hours ago	doc: add API swagger doc for get/post
repositories		8 hours ago	doc: add API swagger doc for get/post
resources		8 hours ago	doc: add API swagger doc for get/post
services		8 hours ago	doc: add API swagger doc for get/post
utils		5 hours ago	Merge branch 'kevin3' of bitbucket.org:poe-harmonic/api_restfull
__init__.py	0 B	12 hours ago	config my test files
app.py	464 B	10 hours ago	Fixed bug about the absolute path for architecture

Structure Test



api_restfull / tests

Name

..

fonctional

integration

unit

__init__.py

conftest.py

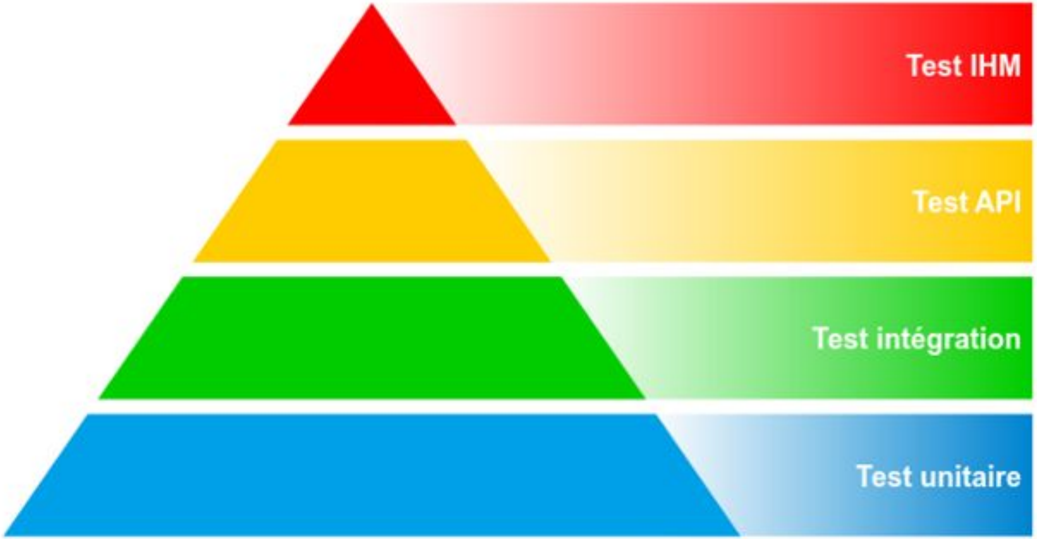
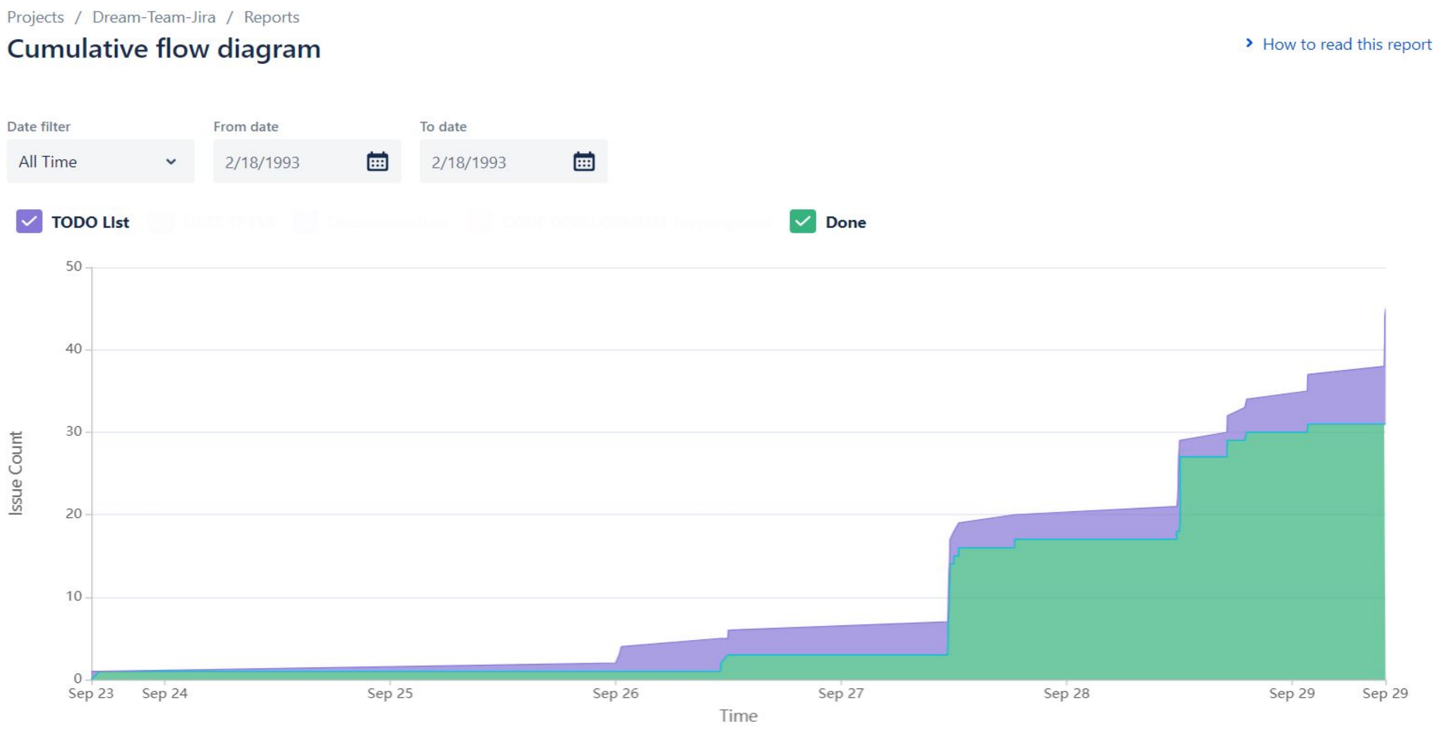


Diagramme de flux cumulatif



Projects / Dream-Team-Jira / Reports

Cumulative flow diagram

› How to read this report

Date filter: All Time

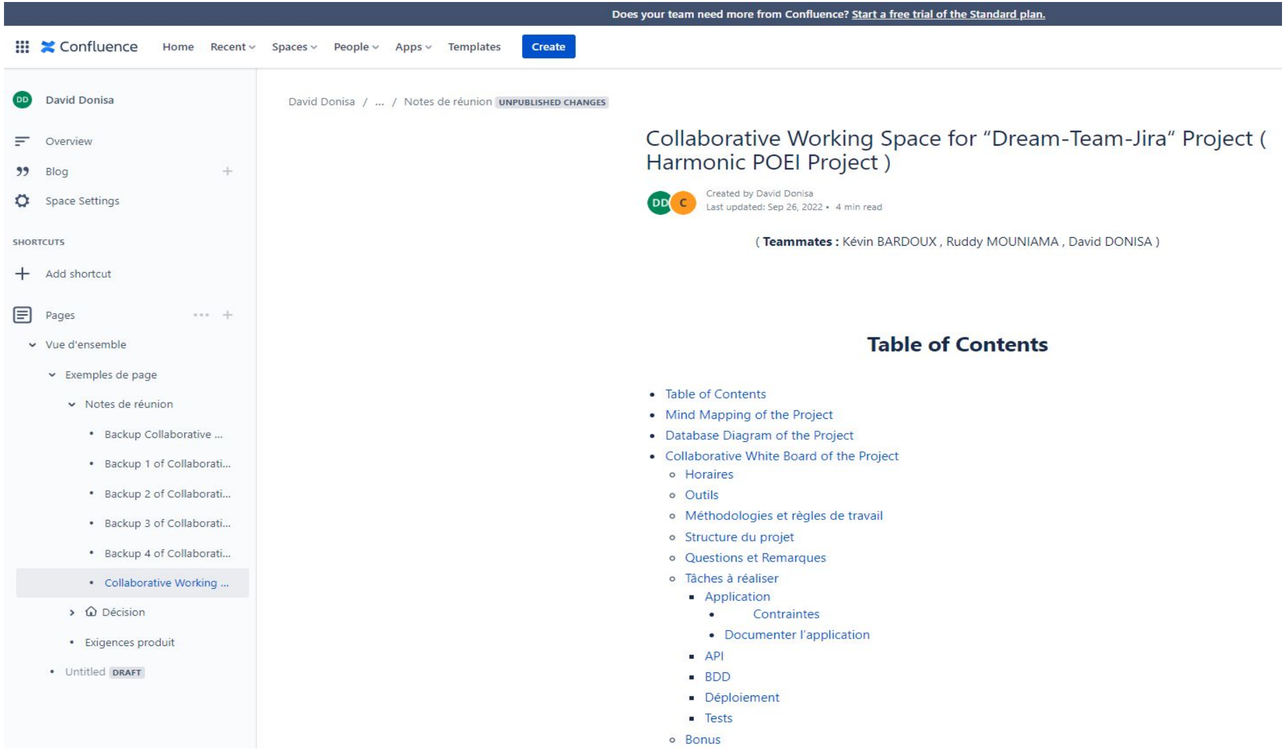
From date: 2/18/1993

To date: 2/18/1993

☑ TODO LIst □ UNIT TESTS □ Documentation □ CODE DEVELOPMENT (in progress) ☑ Done

RÉALISATION

Agilité et Tâches Communes



Does your team need more from Confluence? Start a free trial of the Standard plan.
Confluence
Home
Recent
Spaces
People
Apps
Templates
Create
David Donisa
Overview
Blog
Space Settings
SHORTCUTS
Add shortcut
Pages
Vue d'ensemble
Exemples de page
Notes de réunion
Backup Collaborative ...
Backup 1 of Collaborati...
Backup 2 of Collaborati...
Backup 3 of Collaborati...
Backup 4 of Collaborati...
Collaborative Working ...
Décision
Exigences produit
Untitled DRAFT
David Donisa / ... / Notes de réunion UNPUBLISHED CHANGES

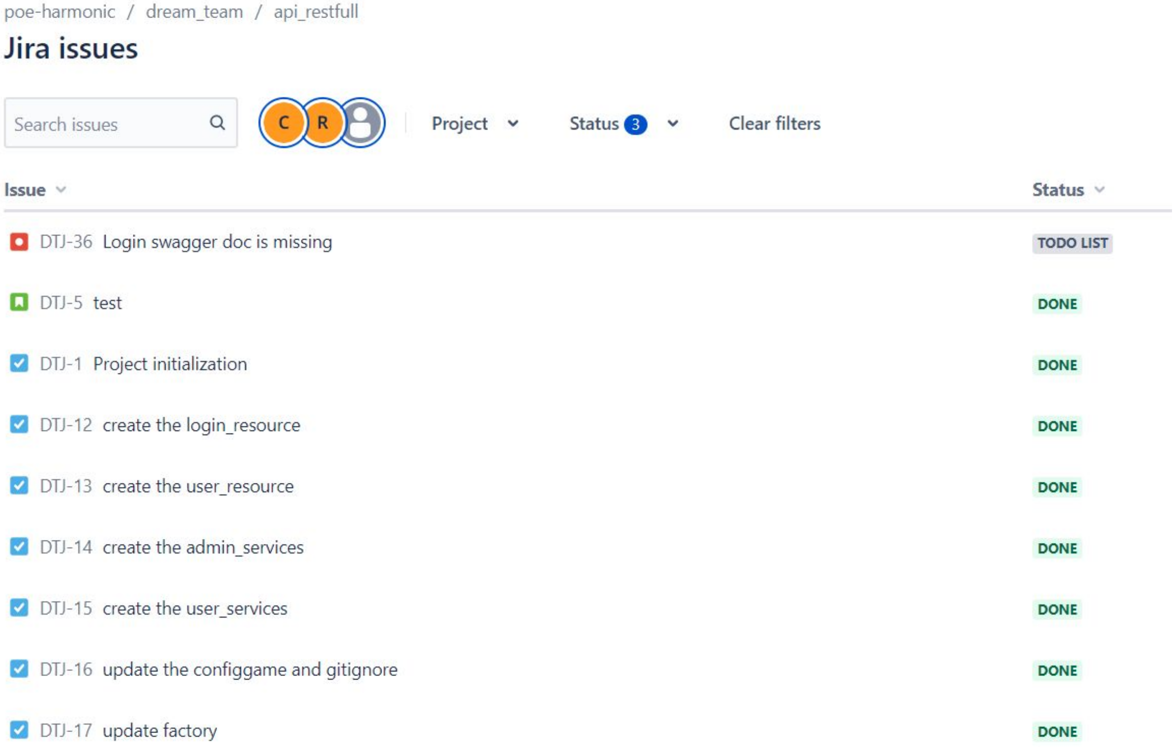
Collaborative Working Space for "Dream-Team-Jira" Project (Harmonic POEI Project)

Created by David Donisa
Last updated: Sep 26, 2022 • 4 min read

(**Teammates :** Kévin BARDOUX , Ruddy MOUNIAMA , David DONISA)

Table of Contents

Tâches

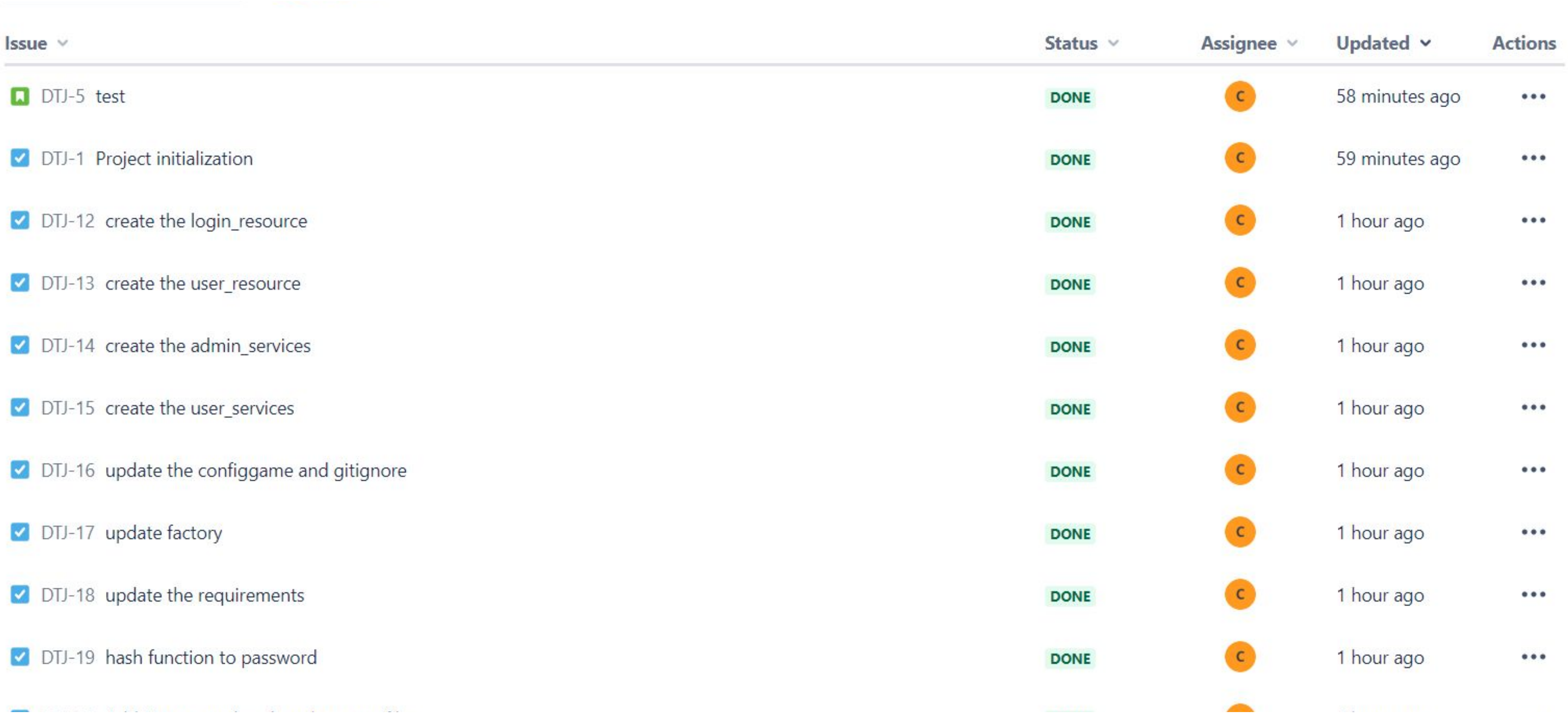


poe-harmonic / dream_team / api_restfull

Jira issues

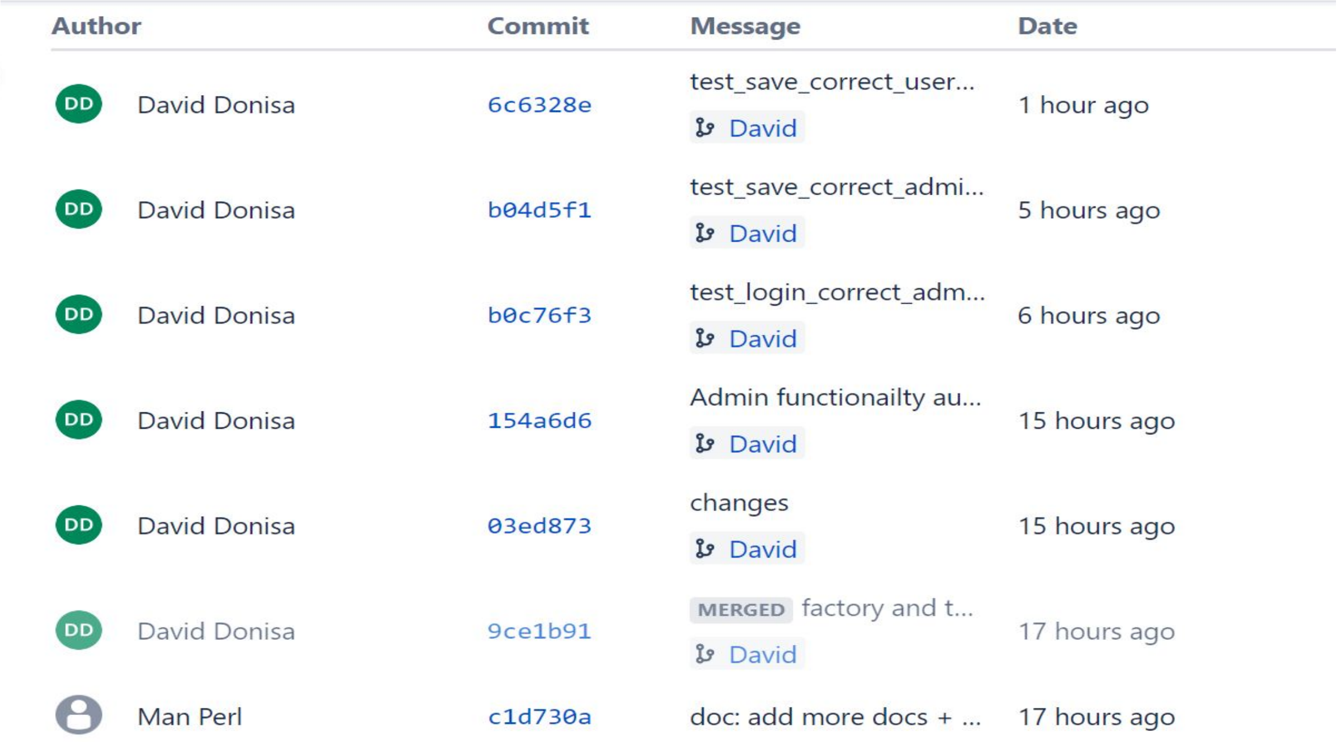
Issue	Status
🟥 DTJ-36 Login swagger doc is missing	TODO LIST
🟩 DTJ-5 test	DONE
☑ DTJ-1 Project initialization	DONE
☑ DTJ-12 create the login_resource	DONE
☑ DTJ-13 create the user_resource	DONE
☑ DTJ-14 create the admin_services	DONE
☑ DTJ-15 create the user_services	DONE
☑ DTJ-16 update the configgame and gitignore	DONE
☑ DTJ-17 update factory	DONE

Tâches Kévin



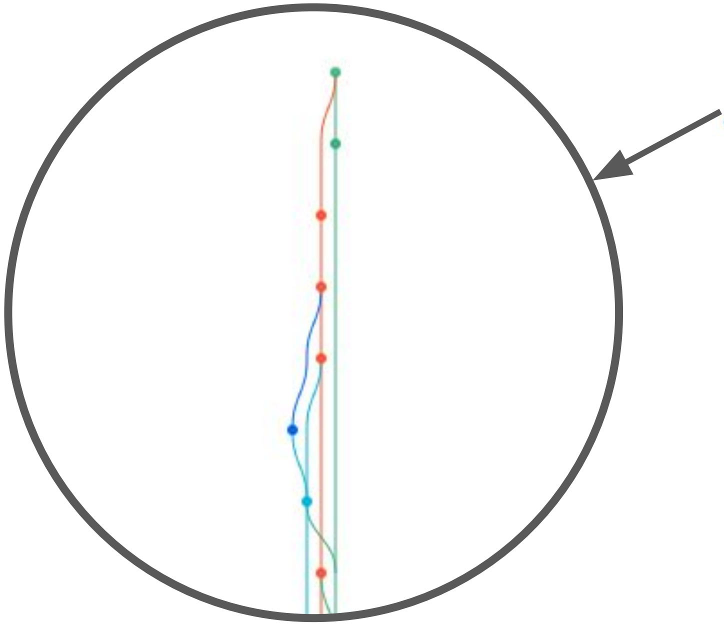
Issue	Status	Assignee	Updated	Actions
🟩 DTJ-5 test	DONE	C	58 minutes ago	•••
☑ DTJ-1 Project initialization	DONE	C	59 minutes ago	•••
☑ DTJ-12 create the login_resource	DONE	C	1 hour ago	•••
☑ DTJ-13 create the user_resource	DONE	C	1 hour ago	•••
☑ DTJ-14 create the admin_services	DONE	C	1 hour ago	•••
☑ DTJ-15 create the user_services	DONE	C	1 hour ago	•••
☑ DTJ-16 update the configgame and gitignore	DONE	C	1 hour ago	•••
☑ DTJ-17 update factory	DONE	C	1 hour ago	•••
☑ DTJ-18 update the requirements	DONE	C	1 hour ago	•••
☑ DTJ-19 hash function to password	DONE	C	1 hour ago	•••

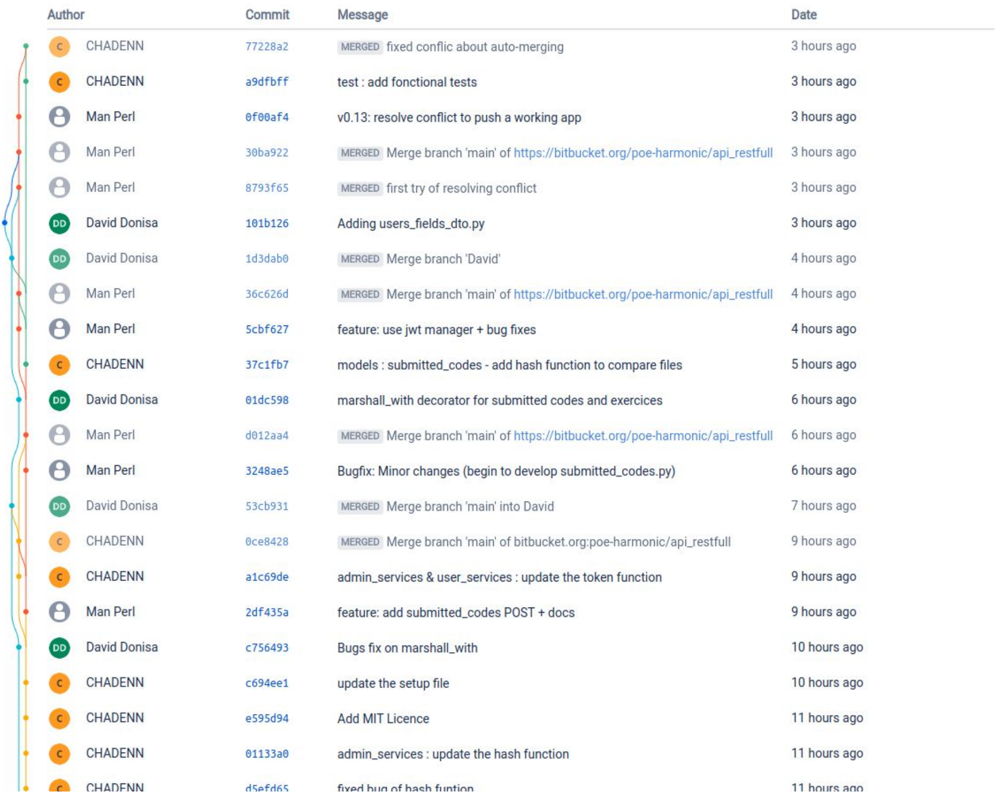
Tâches David



Author	Commit	Message	Date
DD David Donisa	6c6328e	test_save_correct_user... David	1 hour ago
DD David Donisa	b04d5f1	test_save_correct_admi... David	5 hours ago
DD David Donisa	b0c76f3	test_login_correct_adm... David	6 hours ago
DD David Donisa	154a6d6	Admin functionailty au... David	15 hours ago
DD David Donisa	03ed873	changes David	15 hours ago
DD David Donisa	9ce1b91	MERGED factory and t... David	17 hours ago
Man Perl	c1d730a	doc: add more docs + ...	17 hours ago

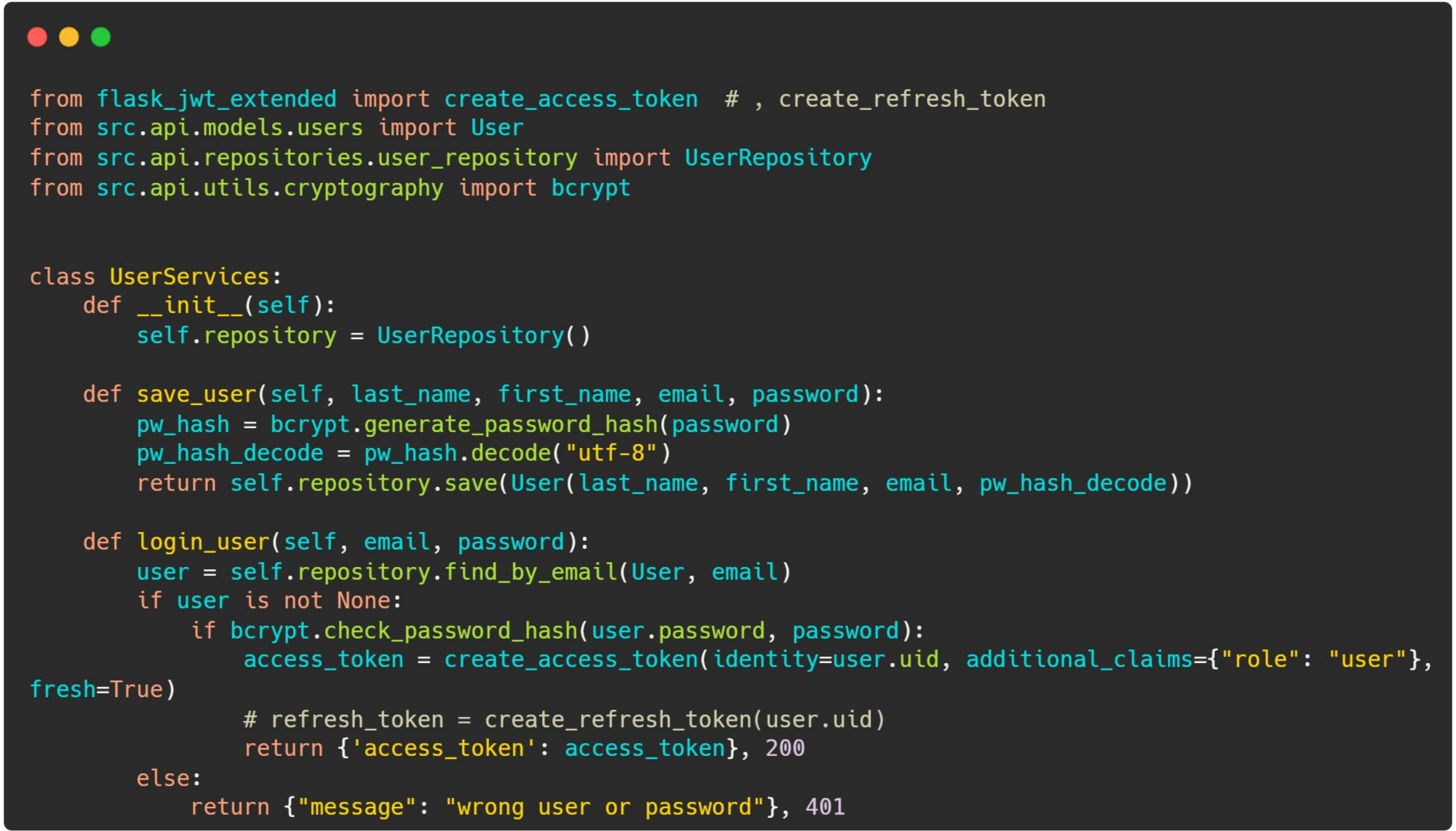
Flow





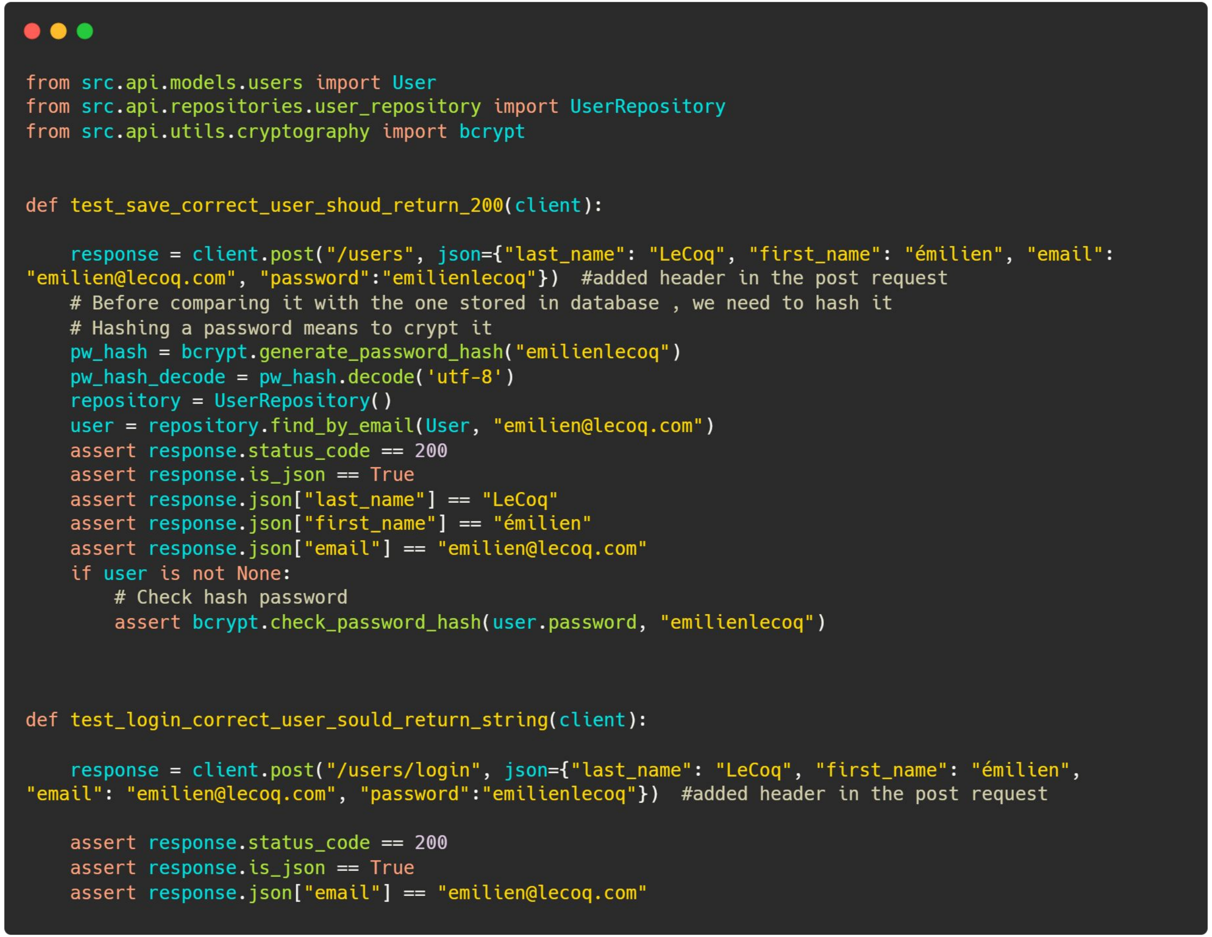
Author	Commit	Message	Date
CHADENN	77228a2	MERGED fixed conflic about auto-merging	3 hours ago
CHADENN	a9dfbff	test : add fonctional tests	3 hours ago
Man Perl	0f00af4	v0.13: resolve conflict to push a working app	3 hours ago
Man Perl	30ba922	MERGED Merge branch 'main' of https://bitbucket.org/poe-harmonic/api_restfull	3 hours ago
Man Perl	8793f65	MERGED first try of resolving conflict	3 hours ago
David Donisa	101b126	Adding users_fields_dto.py	3 hours ago
David Donisa	1d3dab0	MERGED Merge branch 'David'	4 hours ago
Man Perl	36c626d	MERGED Merge branch 'main' of https://bitbucket.org/poe-harmonic/api_restfull	4 hours ago
Man Perl	5cbf627	feature: use jwt manager + bug fixes	4 hours ago
CHADENN	37c1fb7	models : submitted_codes - add hash function to compare files	5 hours ago
David Donisa	01dc598	marshall_with decorator for submitted codes and exercices	6 hours ago
Man Perl	d012aa4	MERGED Merge branch 'main' of https://bitbucket.org/poe-harmonic/api_restfull	6 hours ago
Man Perl	3248ae5	Bugfix: Minor changes (begin to develop submitted_codes.py)	6 hours ago
David Donisa	53cb931	MERGED Merge branch 'main' into David	7 hours ago
CHADENN	0ce8428	MERGED Merge branch 'main' of bitbucket.org:poe-harmonic/api_restfull	9 hours ago
CHADENN	a1c69de	admin_services & user_services : update the token function	9 hours ago
Man Perl	2df435a	feature: add submitted_codes POST + docs	9 hours ago
David Donisa	c756493	Bugs fix on marshall_with	10 hours ago
CHADENN	c694ee1	update the setup file	10 hours ago
CHADENN	e595d94	Add MIT Licence	11 hours ago
CHADENN	01133a0	admin_services : update the hash function	11 hours ago
CHADENN	d5efd65	fixed bug of hash funtion	11 hours ago

Ajoût utilisateur



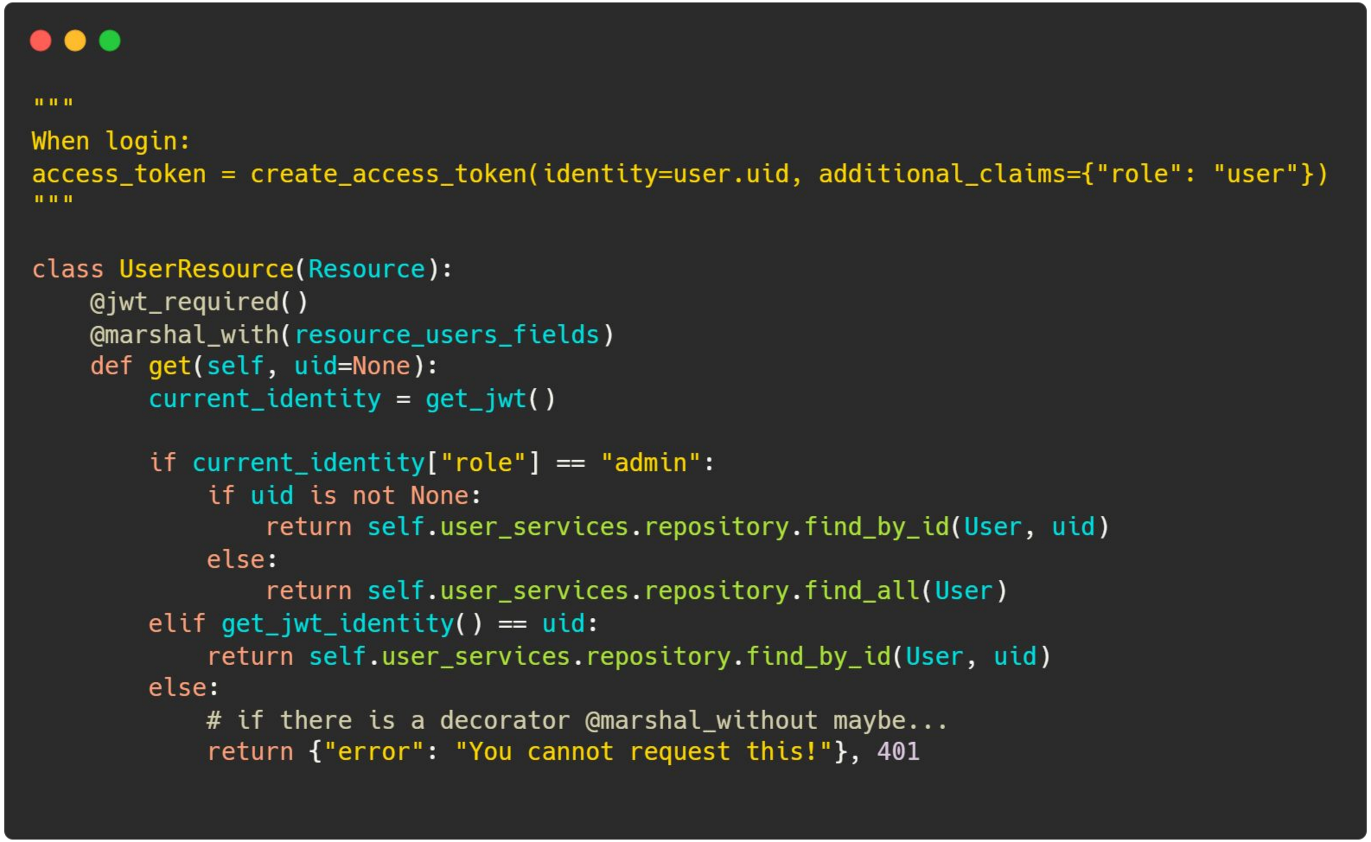
```python
from flask_jwt_extended import create_access_token  # , create_refresh_token
from src.api.models.users import User
from src.api.repositories.user_repository import UserRepository
from src.api.utils.cryptography import bcrypt


class UserServices:
    def __init__(self):
        self.repository = UserRepository()

    def save_user(self, last_name, first_name, email, password):
        pw_hash = bcrypt.generate_password_hash(password)
        pw_hash_decode = pw_hash.decode("utf-8")
        return self.repository.save(User(last_name, first_name, email, pw_hash_decode))

    def login_user(self, email, password):
        user = self.repository.find_by_email(User, email)
        if user is not None:
            if bcrypt.check_password_hash(user.password, password):
                access_token = create_access_token(identity=user.uid, additional_claims={"role": "user"},
fresh=True)
                # refresh_token = create_refresh_token(user.uid)
                return {'access_token': access_token}, 200
        else:
            return {"message": "wrong user or password"}, 401
```

Test Fonctionnel



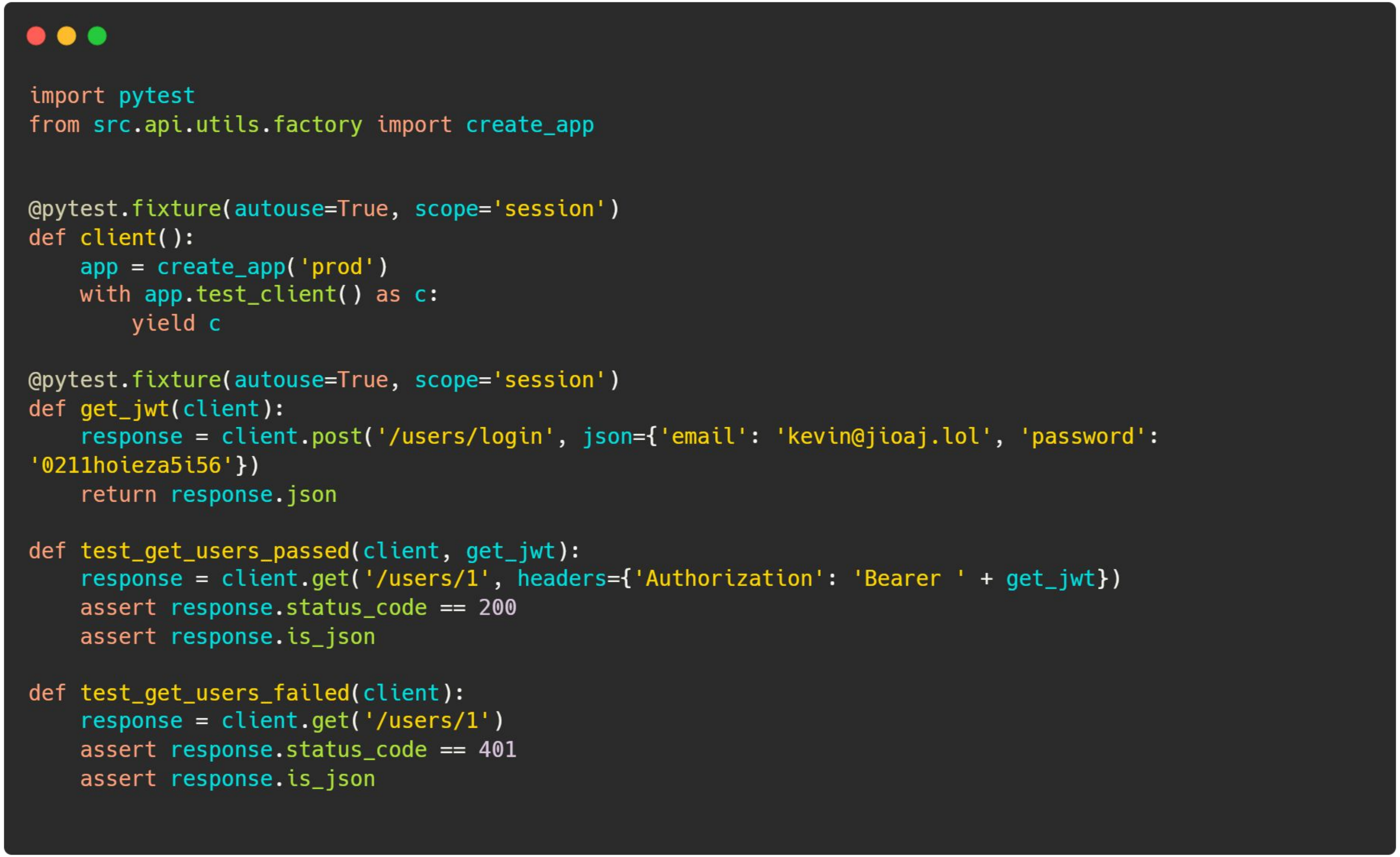
```
from src.api.models.users import User
from src.api.repositories.user_repository import UserRepository
from src.api.utils.cryptography import bcrypt


def test_save_correct_user_shoud_return_200(client):

    response = client.post("/users", json={"last_name": "LeCoq", "first_name": "émilien", "email":
"emilien@lecoq.com", "password":"emilienlecoq"})  #added header in the post request
    # Before comparing it with the one stored in database , we need to hash it
    # Hashing a password means to crypt it
    pw_hash = bcrypt.generate_password_hash("emilienlecoq")
    pw_hash_decode = pw_hash.decode('utf-8')
    repository = UserRepository()
    user = repository.find_by_email(User, "emilien@lecoq.com")
    assert response.status_code == 200
    assert response.is_json == True
    assert response.json["last_name"] == "LeCoq"
    assert response.json["first_name"] == "émilien"
    assert response.json["email"] == "emilien@lecoq.com"
    if user is not None:
        # Check hash password
        assert bcrypt.check_password_hash(user.password, "emilienlecoq")



def test_login_correct_user_sould_return_string(client):

    response = client.post("/users/login", json={"last_name": "LeCoq", "first_name": "émilien",
"email": "emilien@lecoq.com", "password":"emilienlecoq"})  #added header in the post request

    assert response.status_code == 200
    assert response.is_json == True
    assert response.json["email"] == "emilien@lecoq.com"
```

Accéder aux informations



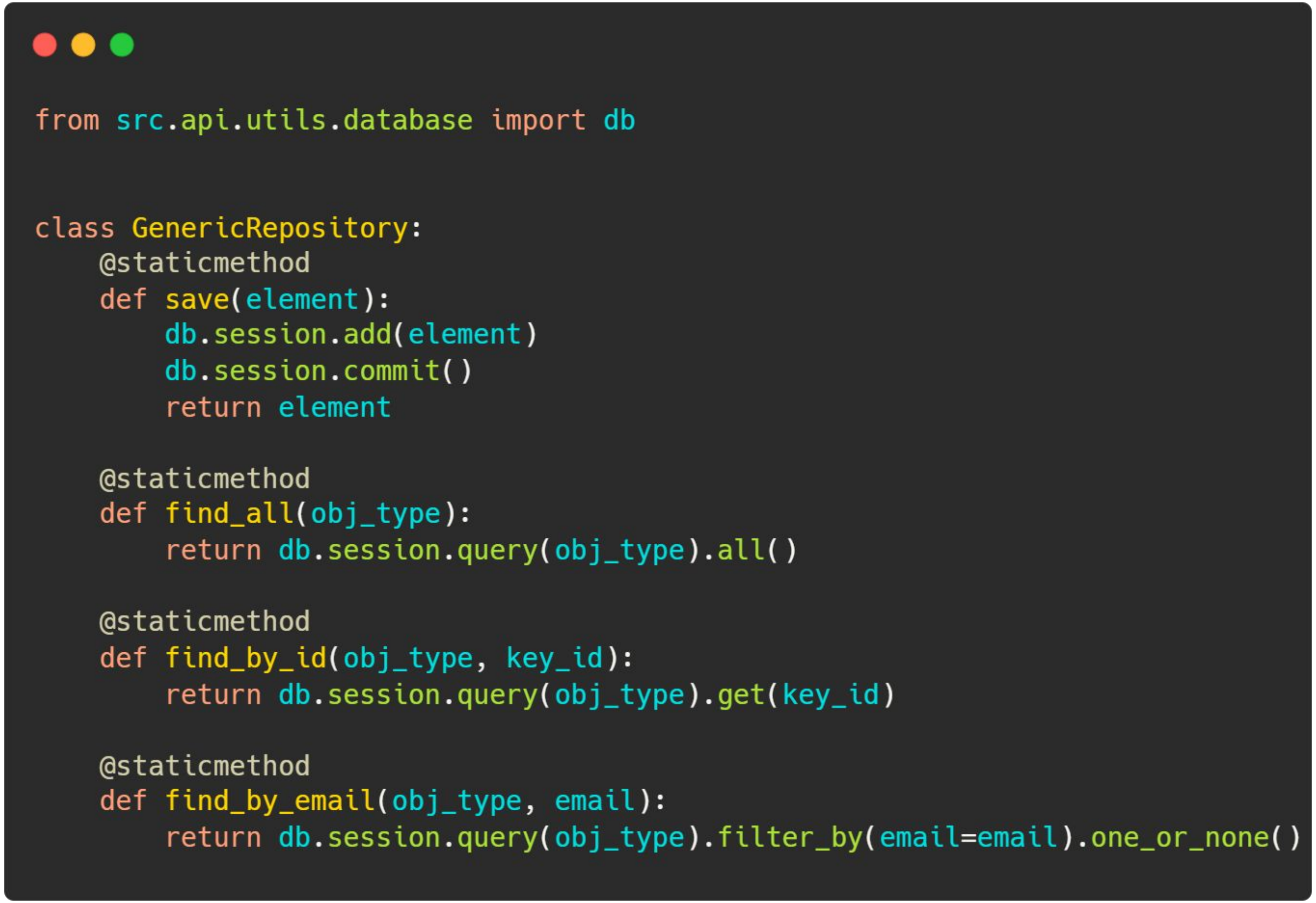
```python
"""
When login:
access_token = create_access_token(identity=user.uid, additional_claims={"role": "user"})
"""

class UserResource(Resource):
    @jwt_required()
    @marshal_with(resource_users_fields)
    def get(self, uid=None):
        current_identity = get_jwt()

        if current_identity["role"] == "admin":
            if uid is not None:
                return self.user_services.repository.find_by_id(User, uid)
            else:
                return self.user_services.repository.find_all(User)
        elif get_jwt_identity() == uid:
            return self.user_services.repository.find_by_id(User, uid)
        else:
            # if there is a decorator @marshal_without maybe...
            return {"error": "You cannot request this!"}, 401
```

Test Fonctionnel



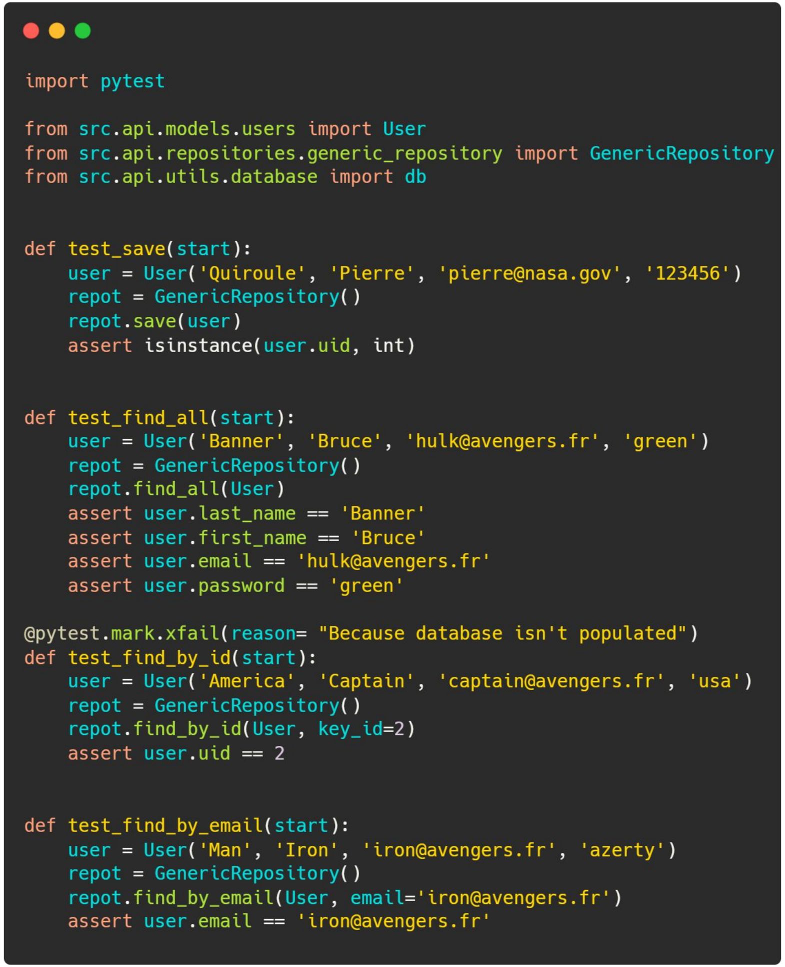
```python
import pytest
from src.api.utils.factory import create_app


@pytest.fixture(autouse=True, scope='session')
def client():
    app = create_app('prod')
    with app.test_client() as c:
        yield c

@pytest.fixture(autouse=True, scope='session')
def get_jwt(client):
    response = client.post('/users/login', json={'email': 'kevin@jioaj.lol', 'password':
'0211hoieza5i56'})
    return response.json

def test_get_users_passed(client, get_jwt):
    response = client.get('/users/1', headers={'Authorization': 'Bearer ' + get_jwt})
    assert response.status_code == 200
    assert response.is_json

def test_get_users_failed(client):
    response = client.get('/users/1')
    assert response.status_code == 401
    assert response.is_json
```

Accès BDD



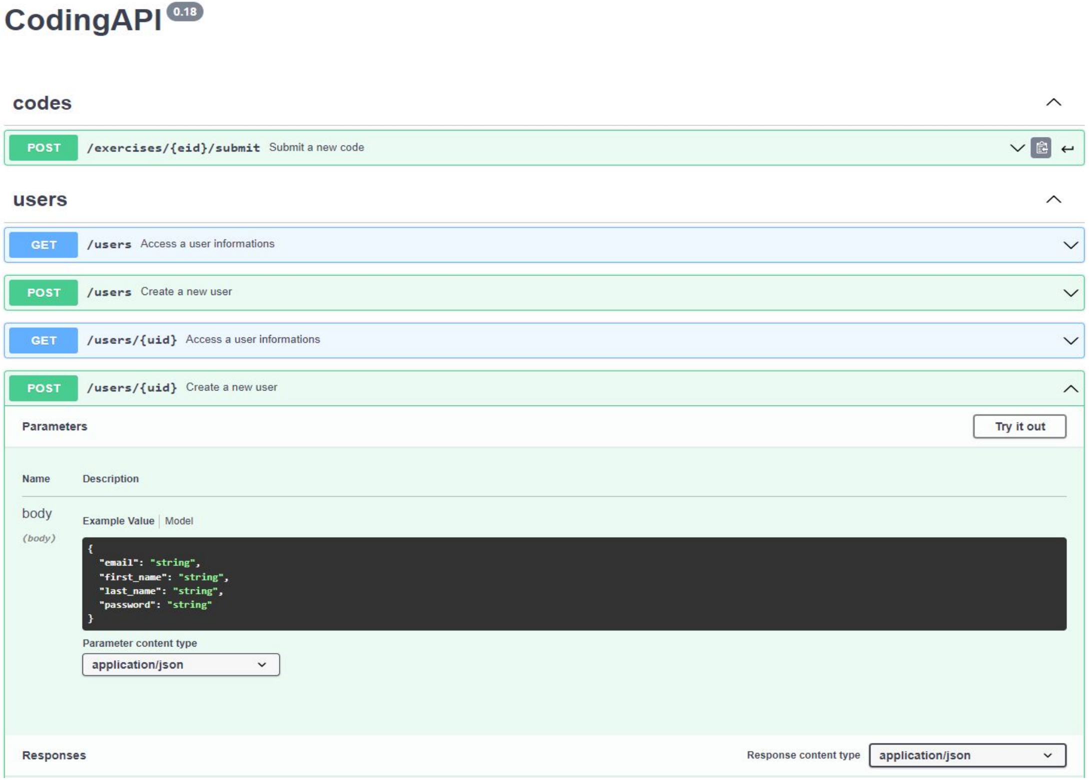
```python
from src.api.utils.database import db


class GenericRepository:
    @staticmethod
    def save(element):
        db.session.add(element)
        db.session.commit()
        return element

    @staticmethod
    def find_all(obj_type):
        return db.session.query(obj_type).all()

    @staticmethod
    def find_by_id(obj_type, key_id):
        return db.session.query(obj_type).get(key_id)

    @staticmethod
    def find_by_email(obj_type, email):
        return db.session.query(obj_type).filter_by(email=email).one_or_none()
```

Test d'intégration



```python
import pytest

from src.api.models.users import User
from src.api.repositories.generic_repository import GenericRepository
from src.api.utils.database import db


def test_save(start):
    user = User('Quiroule', 'Pierre', 'pierre@nasa.gov', '123456')
    repot = GenericRepository()
    repot.save(user)
    assert isinstance(user.uid, int)


def test_find_all(start):
    user = User('Banner', 'Bruce', 'hulk@avengers.fr', 'green')
    repot = GenericRepository()
    repot.find_all(User)
    assert user.last_name == 'Banner'
    assert user.first_name == 'Bruce'
    assert user.email == 'hulk@avengers.fr'
    assert user.password == 'green'

@pytest.mark.xfail(reason= "Because database isn't populated")
def test_find_by_id(start):
    user = User('America', 'Captain', 'captain@avengers.fr', 'usa')
    repot = GenericRepository()
    repot.find_by_id(User, key_id=2)
    assert user.uid == 2


def test_find_by_email(start):
    user = User('Man', 'Iron', 'iron@avengers.fr', 'azerty')
    repot = GenericRepository()
    repot.find_by_email(User, email='iron@avengers.fr')
    assert user.email == 'iron@avengers.fr'
```

Documentation API (swagger)



codes

POST `/exercises/{eid}/submit` Submit a new code

users

GET `/users` Access a user informations

POST `/users` Create a new user

GET `/users/{uid}` Access a user informations

POST `/users/{uid}` Create a new user

body *(body)*

```
{
  "email": "string",
  "first_name": "string",
  "last_name": "string",
  "password": "string"
}
```

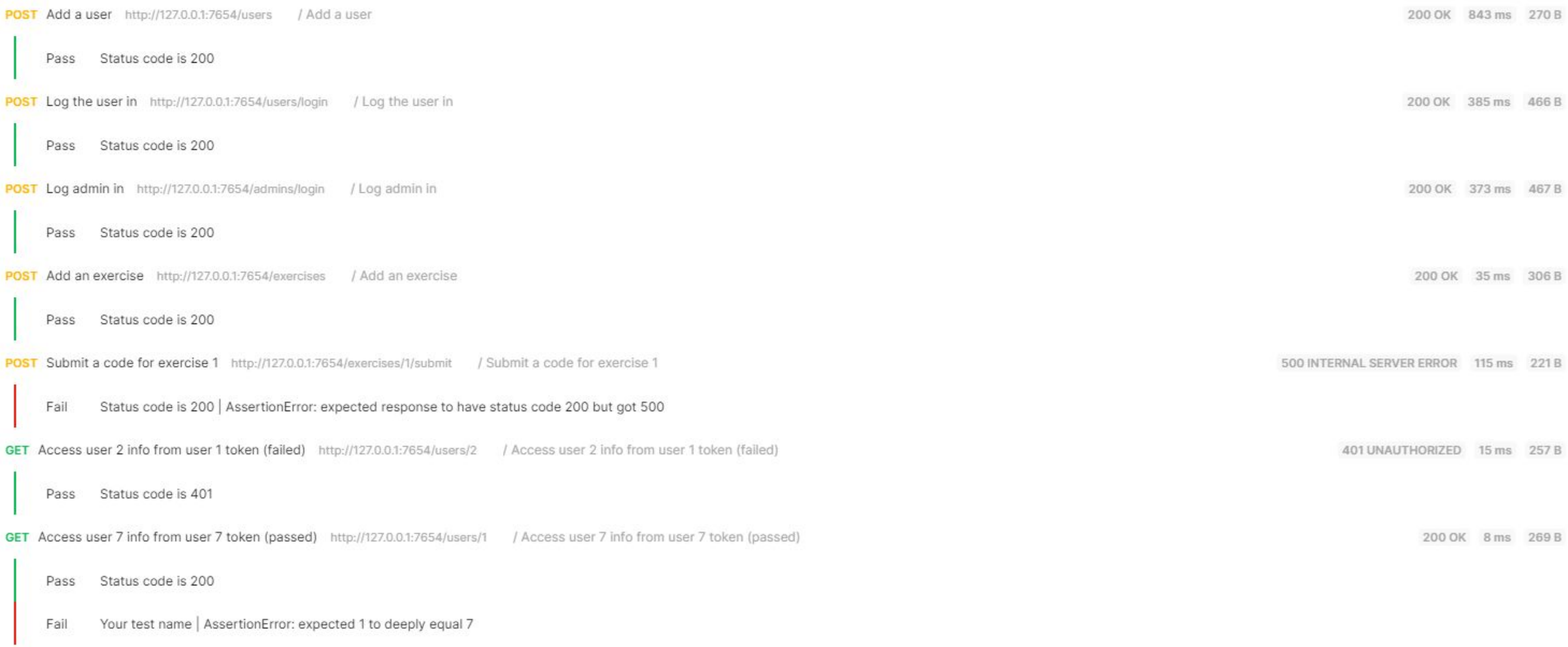
Parameter content type: application/json

Response content type: application/json

DEMO

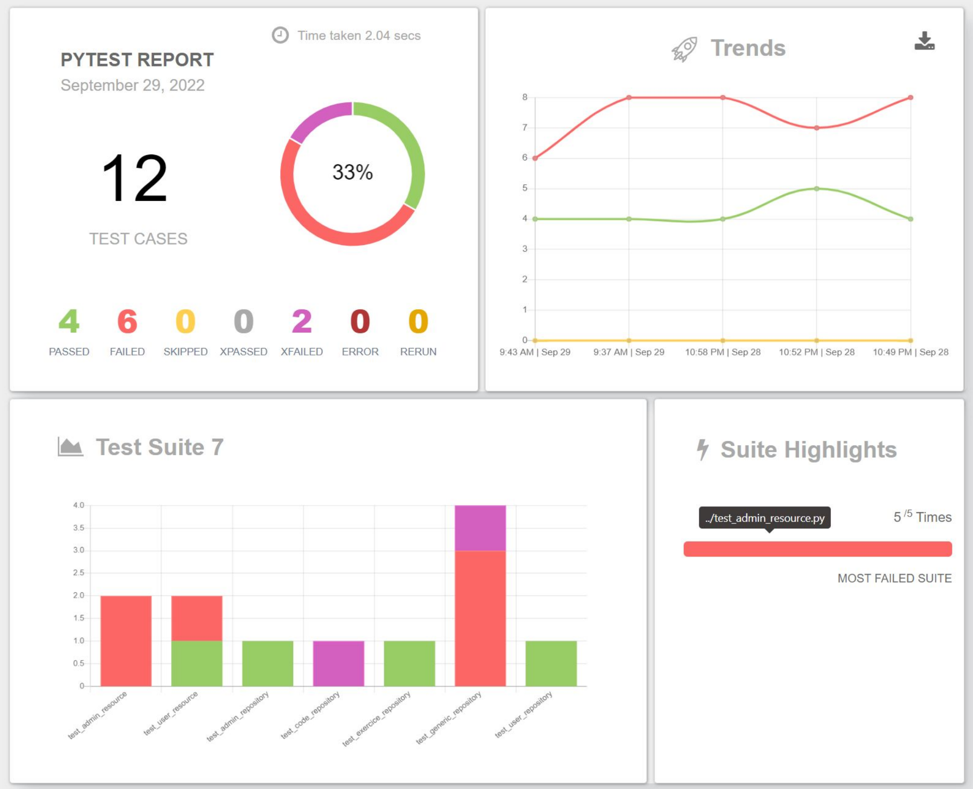
TESTS

Rapport de tests côté client (postman)



```
POST  Add a user   http://127.0.0.1:7654/users   / Add a user                                    200 OK   843 ms   270 B
      Pass    Status code is 200
POST  Log the user in   http://127.0.0.1:7654/users/login   / Log the user in                    200 OK   385 ms   466 B
      Pass    Status code is 200
POST  Log admin in   http://127.0.0.1:7654/admins/login   / Log admin in                         200 OK   373 ms   467 B
      Pass    Status code is 200
POST  Add an exercise   http://127.0.0.1:7654/exercises   / Add an exercise                      200 OK   35 ms   306 B
      Pass    Status code is 200
POST  Submit a code for exercise 1   http://127.0.0.1:7654/exercises/1/submit   / Submit a code for exercise 1   500 INTERNAL SERVER ERROR   115 ms   221 B
      Fail    Status code is 200 | AssertionError: expected response to have status code 200 but got 500
GET   Access user 2 info from user 1 token (failed)   http://127.0.0.1:7654/users/2   / Access user 2 info from user 1 token (failed)   401 UNAUTHORIZED   15 ms   257 B
      Pass    Status code is 401
GET   Access user 7 info from user 7 token (passed)   http://127.0.0.1:7654/users/1   / Access user 7 info from user 7 token (passed)   200 OK   8 ms   269 B
      Pass    Status code is 200
      Fail    Your test name | AssertionError: expected 1 to deeply equal 7
```

Rapport de tests côté server



Time taken 2.04 secs

PYTEST REPORT
September 29, 2022

12
TEST CASES

4	6	0	0	2	0	0
PASSED	FAILED	SKIPPED	XPASSED	XFAILED	ERROR	RERUN

Suite Highlights

../test_admin_resource.py 5 /5 Times

MOST FAILED SUITE

MERCI

QUESTIONS?