# EndPointy

## Djoser:

<https://djoser.readthedocs.io/en/latest/base_endpoints.html>

* auth/users/
* auth/users/me/
  + zwraca podstawowe dane o zalogowanym użytkowniku
* auth/users/confirm/
* auth/users/resend\_activation/
* auth/users/set\_password/
* auth/users/reset\_password/
* auth/users/reset\_password\_confirm/
* auth/users/set\_username/
* auth/users/reset\_username/
* auth/users/reset\_username\_confirm/

https://djoser.readthedocs.io/en/latest/jwt\_endpoints.html

* auth/token/login/ (Token Based Authentication)
  + w sumie nwm
* auth/token/logout/ (Token Based Authentication)
  + POST wylogowywuje token przesłany w nagłówku
* auth/jwt/create/ (JSON Web Token Authentication)
  + POST: po przesłaniu maila i hasła loguje użytkownika
* auth/jwt/refresh/ (JSON Web Token Authentication)
  + Odświeża token
* auth/jwt/verify/ (JSON Web Token Authentication)
  + Use this endpoint to verify JWT

## Moje

* Projects/api/Projects/
  + Bez zalogowania
    - Parametry GET – wyświetla projekty
      * titlecontain=<Co ma zawierać tytuł>
      * sort=<Nazwa pola> //np. pk, title, averageRate// - przed nazwą oznacza odwrotną kolejność np. sort=- averageRate
      * descecontain=<Co ma zawierać opis>
      * up=<górna granica liczby projektów/rekordów>
      * down=<dolna jw. Domyślnie 0>
  + Z zalogowaniem
    - POST - tworzy projekt
      * Np.

{

"title": "tytuł,

"description":"opis"

}

* Projects/api/Projects/<pk>/
  + Bez Logowania
    - GET Zwraca konkretny projekt
  + Po zalogowaniu
    - DELETE usuwa jeśli jest się właścicielem
    - PUT Uaktualnia konkretne dane
* Projects/api/Projects/<pk>/mark
  + POST Pozwala ocenić projekt jeśli użytkownik jest zautoryzowany
    - Np. {„mark”: ”1”}
* Users/api/Users