Group: Data Analysis Towards Abbreviation

API Doc

- /token: Get an authentication token valid for 10 minutes from server using username and password.
  Params: username, password
  return: {"token": HS256_TOKEN} 200 OR 401

- /register: Register a new user with new username, email and password. POST, expect model.
  Params: username, email, password.
  Return: {"message": "Register Successfully"} 200 OR 400

- /user/prefs: Update preferences on different aspect of ratings. POST, expect model, requires_auth.
  Params: cleanliness, location, communication.
  Return: {"message": "Update User Preferences Success"} 200 OR 400

- /user/change_password: Change existing password of user. POST, expect model, requires_auth.
  Params: old_passward, new_password
  Return: {"message": "Password has been updated."} 200 OR 400

- /user/change_email: Change existing email of user. POST, expect model, requires_auth.
  Params: new_email
  Return: {"message": "Email has been updated."} 200 OR 400

- /user/: Get current logged-in user profile. GET, requires_auth
  Return: {username,email,role,cleanliness,location,communication} 200 OR 401 OR 404

- /property/: Add a new property based on different aspects. POST, expect mode, requires_auth.
  Params: host_neighbourhood, city, property_type, room_type
  Return: {"message": "Property {id} has been successfully updated"} 200 OR 400

- /property/<int:id>: Get properties by ID.
  Params: id
  GET. Return: Property Info JSON Object, 200 OR 404
  DELETE. Return: {"message": "Property {id} is removed."} 200 OR 401 OR 404

- /property/<int:id>/price_estimate: Get estimated price for property by estimating similar properties in the neighbourhood. GET.
  Params: id
  Return: {"return": estimated} 200 OR 404

- /property_list/: Get a property list filtered by many different aspects. GET, requires_auth.
  Params: min_price, max_price, suburb, property_type, room_type, accommodates, cleanliness_rating_weight, location_rating_weight, communication_rating_weight, order_by, sorting, page
  Return: {List of up to 10 properties, current page, total page} 200 OR 400

- /property/<int:id>/date_price: Get properties datetime and price by ID. GET, requires_auth.
  Params: id
  Return: {List of (date & price)} 200 OR 404 OR 401

- /property/<int:id>/prediction: Predict price of property in specific date. GET, requires_auth.
  Params: id, date
  Return: {'predicted_price': predicted_price}, 200 OR 401 OR 400