Pokedex

Pokedex REST API that returns Pokemon information.

Token Authentication optional

For token authentication, in every request API has to send JWT as Bearer in header Authorization if API configuration token enable is true.

Authorization: Bearer {{AccessToken}}

Function	Method
Authorization	Bearer Token
Security claims	JSON Web Token (JWT)
JOSE Header Algorithm	HMAC SHA256 (Base64Url)
JWS Payload	Identity (Must Include user identity)

Pokedex API will receive bearer token as JWT and will decode JWT using a secret key and will make sure that token is valid. In case of an expired or invalid JWT, Pokedex API will return an error with HTTP code 401 (Unauthorized).

API Status Codes

Http Code	Error Code	Description
200	OK	Success
401	INVALID_ACCESS_TOKEN	Invalid Token
		Token is Expired
		User Identity is not present inside JWT payload
		JWT algorithm is not HMAC SHA256
	UNAUTHORIZED	The credentials provided aren't valid.
404	NOT_FOUND	The requested resource does not exist on the server
422	EMPTY_POKEMON_NAME	Pokemon name is empty
500	INTERNAL_SERVER_ERROR	Casting or divide by zero etc problem
502	DATABASE_CONFIGURATION_ERROR	MongoDB configuration error.

API 1 Basic Pokemon Information

/HTTP/GET/pokemon/<pokemon name>
Given a Pokemon name, returns standard Pokemon description and additional information.

Endpoint

http://localhost:8000/pokemon/hasan

Path Variables

#	Field Name	Format	M/O/C	Description
1	name	string	M	Pokemon's name.

Response Data Fields

#	Field Name	Format	M/O/C	Description
1	name	string	M	Pokemon's name.
2	description	string	M	Pokemon's standard description.
3	habitat	string	M	Pokemon's habitat.
4	isLegendary	bool	M	Pokemon's legendary status.

JSON Request and Response Body

Method	Request Path	Response Body	Format

GET
Success
HTTP
200 OK
	http://localhost:8000/pokemon/hasan	{
		"name": "Hasan",
		"description": "It was created by a scientist after years of horrific gene splicing and DNA engineering experiments",
		"habitat": "rare",
		"isLegendary": false
		}
			JSON

GET
Failed
401
	http://localhost:8000/pokemon/hasan1	{
		"error_code": "UNAUTHORIZED",
		"data": {
		"field": "Name",
		"message": "The credentials provided aren't valid."
		}
		}
			JSON

API 2 Translated Pokemon Information

/HTTP/GET/pokemon/translated/<pokemon name>

Given a Pokemon name, returns translated Pokemon description and other basic information using the following rules -

1. If the Pokemon's habitat is a cave or it's a legendary Pokemon then will be applied to Yoda translation.
2. For all other Pokemon will be applied to the Shakespeare translation.
3. If you can't translate the Pokemon's description then it will be used as the standard description.

Endpoint

http://localhost:8000/pokemon/translated/Craig

Path Variables

#	Field Name	Format	M/O/C	Description
1	name	string	M	Pokemon name.

Response Data Fields

#	Field Name	Format	M/O/C	Description
1	name	string	M	Pokemon's name.
2	description	string	M	Pokemon's translated description.
3	habitat	string	M	Pokemon's habitat.
4	isLegendary	bool	M	Pokemon's legendary status.

JSON Request and Response Body

Method	Request Path	Response Body	Format
GET Success HTTP 200 OK	http://localhost:8000/pokemon/translated/Craig	{ "name": "Craig", "description": "Created by a scientist after years of horrific genesplicing and dna engineering experiments, it was", "habitat": "rare", "isLegendary": true }	JSON
GET Failed 401	http://localhost:8000/pokemon/translated/Craig1	{ "error_code": "UNAUTHORIZED", "data": { "field": "Name", "message": "The credentials provided aren't valid." } }	JSON