IGME-430 Project 1 Documentation

[Heroku](https://igme-430-proj-1-team-builder.herokuapp.com/) | [GitHub](https://github.com/HauntedPineapple/IGME-430-Project1) | [CircleCI](https://app.circleci.com/pipelines/github/HauntedPineapple/IGME-430-Project1/4/workflows/5ccef7d5-43c0-49ef-a211-1af0c32b2ebc/jobs/4)

# Main Questions

## What is the purpose of your application?

Allows a user to create a team of pokemon and store that data to be referred to later for any kind of use.

## What data is being stored and delivered by the API?

Team data is being stored and delivered. This consists of an array of [Team](#_Team) objects which has a name, a size and an array containing the pokemon in the team. The pokemon array is populated by [Pokemon](#_Pokemon) objects.

## What went right in the development of this project?

API functions work mostly as expected.

## What went wrong in the development of this project?

Unexpected timing delays kept the project from being completed in the expected timeframe.

## What did you learn while developing this project?

* Node development practices
* How to create an API that works with an existing API

## If you were to continue, what would you do to improve your application?

* Improve frontend with better styling
* Add more inputs to store and get data with
  + Allow user to create tags to attach to created teams to improve organization
  + Allow user to search their teams and filter with tags
* Allow user to edit existing teams

## If you went above and beyond, how did you do so?

## If you used any borrowed code or code fragments, where did you get them from?

Used code from previous assignments.

## What do the code fragments do? Where are they in your code?

Code from previous assignments are used in most of the js files

# Endpoint Documentation

## /createTeam

* Description of what the endpoint does and returns
* Supported Methods:
  + GET
  + HEAD
  + POST
* Query/Body Parameters:

## /getTeams

* Description of what the endpoint does and returns
* Supported Methods:
  + GET
  + HEAD
  + POST
* Query/Body Parameters:

## <Endpoint URL>

* Description of what the endpoint does and returns
* Supported Methods:
  + GET
  + HEAD
  + POST
* Query/Body Parameters:

# Other Notes

## Object Type Definitions

### BaseStats

* hp: *number*
* speed: *number*
* attack: *number*
* special\_attack: *number*
* defense: *number*
* special\_defense: *number*

### Pokemon

* id: *number*
* name: *string*
* types: *string[]*
* baseStats: *BaseStats*
* apiURL: *string*
* spriteURL: *string*

### Team

* name: *string*
* size: *number*
* pokemon: *Pokemon[]*