

Pikachu Bot

Events:

Class **guildMemberAdd**

Adds a new Discord member to the User database when they join the server.

Class **guildMemberRemove**

Removes a Discord member to the User database when they leave the server.

Commands:

NOTE: `<>` represents placeholder name inputs (i.e. replace with actual value that the placeholder represents), whereas `[]` represents optional inputs.

!want (or **!veux**)

- **Usage:**

1. **!want** [`<pokemon>`] [`<neighbourhood>`] [`<level>`] [`<iv>`]

- **Description:** Add preference to database for user

- `<pokemon>`: **all** or a Pokemon name (EN or FR)

- None specified: **all**

- `<neighbourhood>`:

everywhere/locations/location/neighbourhood name or alias

- None specified: **locations** value unless empty, if so then it uses **everywhere**

- **locations**: **locations** value at time of this

- command (**everywhere** if none set)
 - **location**: uses default locations settings dynamically (shows up as **location** but takes on values present inside **Default Locations Defaults**: ... at spawn time)
 - **<level>**: a number **0** to **40**
 - None specified: **0** (i.e. any level)
 - **<iv>**: a number **0**, or **41** to **100**
 - None specified: **0** (i.e. any iv)
 - 2. **!want blacklist <pokemon1[, pokemon2, pokemon3, ...]>**
 - **Description**: Blocks any alerts coming from specified names, useful when having **all** for a preference but wish to ignore common Pokemon
-

!unwant (or **!veuxpas**)

- **Usage**:
 1. **!unwant [<pokemon>] [<neighbourhood>] [<level>] [<iv>]**
 - **Description**: Removes wild preference for user from database
 2. **!unwant blacklist <pokemon1[, pokemon2, pokemon3, ...]>**
 - **Description**: Remove Pokemon from blacklist
-

!locations (or **!location**)

- **Usage**:
 1. **!locations**
 - **Description**: Get user's current set favorites regions
 2. **!locations <neighbourhood1[, neighbourhood2[, ...]]>**
 - **Description**: Set user's current favorites region to every neighbourhood supplied
-

!neighbourhoods (or !quartiers, !neighbourhood, !quartier, !districts, !areas, !sectors, !arrondissements)

- **Usage:**
 1. **!neighbourhoods**
 - **Description:** Prints all avail. neighbourhoods
 2. **!neighbourhoods <firstLetter>**
 - **Description:** Prints all avail. neighbourhoods starting in **<firstLetter>**
-

!translate (or !traduit)

- **Usage:**
 1. **!translate <name>**
 - **Description:** Prints translated Pokemon **<name>**
-

!spell (or !ecrire, !epeler, !write, !name, !pronounce)

- **Usage:**
 1. **!spell <word>**
 - **Description:** Prints possibly corrected **<word>**
 - **<word>**: either a Pokemon name (EN or FR) or neighbourhood name or alias

Monitors:

Class **wild-income**

Relates wild spawn alerts by DM to respective users that match the query matching one of their preferences in the User database.

Notes:

- **TO-DO:** (**guildMemberAdd.js** & **guildMemberRemove.js**)

- ☐ Test scenarios!
- **TO-DO:** (**User.js**)
 - ☒ Find way to make a validator function for each test in order to output error message for each when they fail
- **TO-DO** (**want.run**)
 - ☒ Duplicate db entries
 - ☐ If one entry with same **name**, **iv**, and **level** exists but does not contain one or all of same **neighbourhoods** content, append it to existing **neighbourhoods** array in db
 - ☒ Use of **location** or **locations** as neighbourhood sets the query to his current settings (if any)
- **TO-DO:** (**want.parseWildPreferenceQueryFromArgs**)
 - ☒ Bad entries (out of range for **iv** and **level**)
 - ☒ Bad entries (omitted for **iv** and **level**)
- **TO-DO:** (**unwant.run**)
 - ☒ Remove entry with identical object query
 - ☐ If one entry with same **name**, **iv**, and **level** exists but differs in **neighbourhoods** content, overwrite field with the difference between the old and new one to existing **neighbourhoods** array in db