

# ##### POKENAME GENERATOR #####

[Live version](#)

## What is done

### ## Requirements

- Generates unique version names
- Reads data from the Pokemon API
- Displays table of previously generated version names
- Can remove generated version names
- Uses Backbone JS framework

### ## Bonus Points

- Uses CoffeeScript ,SASS(scss)
- Doesn't use Bootstrap or similar
- Stored names are persisted after page reload
- Offers option to copy generated name to clipboard
- A runnable unit test suite (jsTestDriver)

### ## Extra

- Offers option to generate items sequentially (default) or randomly
- Can edit the fields of the stored item (By clicking on them)
- Can copy to clipboard any of the previous stored names directly

## UI explanation

Upper part (left to right):

- **Generation Mode checkbox**, this toggles between sequential and random generation
- **Version number input**, this can be entered manually (filtered to only allow numbers and dot) but is also generated sequentially after each version generation
- **Generate button**, generated the poke-name for current version (also works with enter), each new item is added to the top of the table (with green color)
- **Sort table button**, sort table based on the name field (also happens every time you reload the page)
- **Clear table button**, removes the entire collection (from both the UI and storage)

Lower part (table):

- **Version, Adjective, Name, Nature**, each of those can be edited by clicking it
- **Remove**, removes this item from the UI and the Collection
- **Copy**, copies Adjective and Name to clipboard

## Functionality

- When the page loads, the entire pokedex is fetched from the pokemon api. This offers the ability to sort the pokemon and ensure that names are generated alphabetically (assuming increasing version numbers).  
In comparison, using an api call on each generation (which I did on earlier versions), not only prevents alphabetical selection, but also fetches a large extra amount of parameters for each pokemon which are not needed (pokedex only fetches name and a uri to it's resource)
- On random mode, both the pokemon name and adjective are picked randomly and the version serves only a cosmetic role
- Name generation happens as follows. Version is split by dot character, first part corresponds to the name of the pokemon, while second part generates the adjective.  
That means that pokemon names, change only on major updates! (which makes sense)  
On each generation a pokemon nature is also randomly picked (just cosmetic).
- The adjective is picked by a compiled list of 200 sorted adjectives which was generated with an online tool, nature is handled similarly (just not sorted).
- When a stored poke-name item is edited, it turns green to let you identify it better, and also tell you that it might be out of sort now (like the newly generated ones)
- There is a resize function that adds compact classes on some elements (on window resize), to ensure the app can be displayed properly on mobile (feel free to resize the browser to see)

## Notes

(just some things I'd like to mention regarding development)

The clipboard functionality is actually applied on mouse hover over the corresponding button. I arrived to that solution after trying to apply it on each item of the table during generation and found that it greatly impedes performance as the collection grows larger.

I also ensured that previous flash embeds were destroyed before adding the next one to prevent entropy after a long time of continuous usage

I decided to skip jade since I'm using emmet plugin on netbeans which offers me on demand templating (not so strong I guess but still plenty useful)

Regarding unit testing, I decided to go with jsTestDriver, unfortunately as I found out later it doesn't offer me the ability to test DOM related parts, so I only made tests for some (still important) utility functions

Copy to clipboard doesn't work if you run the project using [file:///](#) protocol, this is due to flash security restrictions

There are more things to mention, but they may be better left for discussing (this is already too long!!!)